

WO 2005087452 A1

2005JP2005004838 2005/ 3/17

JP 2004 2004077214 A (2004/ 3/17)

2005/ 9/22

[ENG] ROBOT DEVICE, BEHAVIOR CONTROL METHOD FOR THE ROBOT DEVICE,  
AND MOVING DEVICE

[FRE] DISPOSITIF ROBOT, PROCEDE DE COMMANDE DE COMPORTEMENT  
POUR CE DISPOSITIF ROBOT ET DISPOSITIF MOBILE

SONY CORP (JP); GUTMANN STEFFEN (JP); FUKUCHI MASAKI (JP)

GUTMANN STEFFEN (JP); FUKUCHI MASAKI (JP)

B 25J 13/00 A; B 25J 5/00 B

[ECLA] B62D57/032

AE AG AL AM AT AU AZ BA BB BG BR BW BY BZ CA CH  
CN CO CR CU CZ DE DK DM DZ EC EE EG ES FI GB GD  
GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC  
LK LR LS LT LU LV MA MD MG MK MN MW MX MZ NA NI  
NO NZ OM PG PH PL PT RO RU SC SD SE SG SK SL SM  
SY TJ TM TN TR TT TZ UA UG US UZ VC VN YU ZA ZM  
ZW [BW GH GM KE LS MW MZ NA SD SL SZ TZ UG ZM  
ZW AM AZ BY KG KZ MD RU TJ TM AT BE BG CH CY CZ  
DE DK EE ES FI FR GB GR HU IE IS IT LT LU MC NL PL  
PT RO SE SI SK TR BF BJ CF CG CI CM GA GN GQ GW  
ML MR NE SN TD TG]

abstract A robot device observes the external world by a stereo vision system (1)  
and outputs stereo data (D1) as a distance image, the stereo data (D1) being  
three-dimensional distance information calculated by a parallax of both eyes. A plane

detector (2) detects planes from the distance image to recognize the planes present in the environment. From plane data (D2), a stair recognizer (3) extracts a plane which the robot can climb up and down, recognizes a stair from that plane, and outputs stair data (D4). A stair climb-up/-down controller (4) outputs a behavior control command (D5) for realizing the movement for climbing up and down the stair by using the stair data (D4). This enables a mobile body itself to obtain information on a stair to autonomously perform the movement of stair climb up and down.

Dispositif robot qui observe le monde extérieur à l'aide d'un système de vision stéréo (1) et produit des données stéréo (D1) en tant qu'image de distance, lesdites données stéréo (D1) étant des informations de distance en trois dimensions calculées par une parallaxe des deux yeux. Un détecteur (2) de plans détecte des plans à partir de l'image de distance pour reconnaître les plans présents dans l'environnement. A partir des données de plans (D2), un reconnaisseur (3) d'escalier extrait un plan que le robot peut monter et descendre, reconnaît un escalier dans ce plan et produit des données d'escalier (D4). Un dispositif de commande de montée / descente (4) des marches produit une instruction de commande de comportement (D5) permettant au robot d'effectuer le mouvement de montée et de descente des marches à l'aide des données d'escalier (D4). Cela permet à un corps mobile d'obtenir des informations sur un escalier pour effectuer de manière autonome le mouvement de montée et de descente dudit escalier.

-----  
-----  
2005/ 9/22 AK + DESIGNATED STATES

(A1) AE AG AL AM AT AU AZ BA BB BG BR BW BY BZ CA CH CN CO CR CU CZ  
DE DK DM DZ EC EE EG ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR  
KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ NA NI NO NZ OM PG PH  
PL PT RO RU SC SD SE SG SK SL SM SY TJ TM TN TR TT TZ UA UG US UZ VC VN  
YU ZA ZM ZW

2005/ 9/22 AL + DESIGNATED COUNTRIES FOR REGIONAL PATENTS

(A1) GM KE LS MW MZ NA SD SL SZ TZ UG ZM ZW AM AZ BY KG KZ MD RU TJ

TM AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IS IT LT LU MC NL PL PT  
RO SE SI SK TR BF BJ CF CG CI CM GA GN GQ GW ML MR NE SN TD TG

2005/11/16/ 121 EP: THE EPO HAS BEEN INFORMED BY WIPO THAT EP WAS  
DESIGNATED IN THIS APPLICATION

2006/ 9/14/ ENP ENTRY INTO THE NATIONAL PHASE IN:

2006511065 JP (A)

## DESCRIPTION

### Robot Apparatus, Method of controlling the motion thereof, and Locomotion Mechanism

#### Technical Field

The present invention relates to a robot apparatus, a locomotion mechanism used in the robot apparatus and including locomotion means such as legs etc. to enable the robot apparatus to climb a stairway plural steps and to move otherwise, and a method of controlling the stair climbing of the robot apparatus.

This application claims the priority of the Japanese Patent Application No. 2004-077214 filed on March 17, 2004, the entirety of which is incorporated by reference herein.

#### Background Art

Heretofore, multiple techniques have been proposed for enabling a robot apparatus with a locomotion mechanism including locomotion means such as wheels and the like to move in a stepped environment and up and down on stairs.

The above conventional techniques include, for example, a method of enabling a robot apparatus to move up and down on an object on the basis of already-known stair information including object node data intended for recognition of shape features of the object as shown in FIG. 1 (as disclosed in the



Japanese Patent No. 3176701) and a method of enabling a robot apparatus to move up and down on stairs on the basis of stair data detected by many tactile sensors 602 attached to an entire foot base 601 in the form of a matrix and covered with a protective membrane 603 as shown in FIG. 2 (as disclosed in the Japanese Patent No. 3278467). The techniques also include a method of enabling the stair climbing of a robot apparatus by providing an infrared sensor on either side of each foot base of the robot apparatus and a landmark tape on either side of the stair as shown in FIG. 3 (as disclosed in the Japanese Patent No. 3330710). In the above third method (disclosed in the Japanese Patent No. 3330710), there are used a plurality of photosensors 682 provided on either side of each of foot bases 622R and 622L of a bipedal walking (or biped locomotion) robot apparatus, and straight landmarks 680 each formed from a paint which will well absorb light, such as black paint, and having a predetermined width as shown in FIG. 3A. Outputs from the photosensors 682 in pair are compared with each other to detect a direction of the feet in relation to the landmarks 680. With such foot bases 622R and 622L and landmarks 680, it is possible to recognize the position on a stair 690 as shown in FIG. 3B.

However, since the robot apparatus is controlled to move up and down on stairs on the basis of the already-known stair information, the technique disclosed in the Japanese Patent No. 3176701 is not usable to control the movement of the robot apparatus in any unknown environment. Therefore, it is difficult to apply

this technique to any autonomous robot apparatus etc. for example. Also, the techniques disclosed in the Japanese Patent Nos. 3278467 and 3330710 are such that the stair climbing of the robot apparatus is controlled based on the stair information detected by the plurality of photosensors provided on or on either side of the foot base. Thus, the stair information cannot be acquired before the robot apparatus places the feet on the tread of a stair. Since no details of any stairs can be viewed from a far position, these techniques can only be utilized in case it can be predicted that stairs exist ahead of the robot apparatus.

#### Disclosure of the Invention

It is therefore an object of the present invention to overcome the above-mentioned drawbacks of the related art by providing a robot apparatus, a locomotion mechanism that enables autonomous stair climbing of the robot apparatus by acquiring information on stairs on which the robot apparatus is going to move up or down, and a method of controlling the motion of the robot apparatus.

The above object can be attained by providing a robot apparatus enabled by a locomotive means to move, the apparatus including according to the present invention:

a plane segmenting/extracting means for detecting a plane or a plurality of planes included in an environment surrounding the apparatus from three-dimensional distance data and outputting the detected plane or planes as plane

information;

a stair recognizing means for recognizing, based on the plane information, a stair having a plane on which the apparatus can move and outputting stair information including information on the tread and riser of the stair; and

a stair-climb controlling means for controlling the stair climbing of the apparatus by judging, based on the stair information, whether the apparatus can climb the stair and having the apparatus autonomously position itself in relation to the stair tread when the stair is determined to be climbable.

In the above locomotion robot apparatus including legs, for example, as the locomotion means according to the present invention, it is judged, based on the tread information such as the dimensions, positions, etc. of a tread, whether the foot bases of the apparatus can be placed on the tread, and it is judged, based on the riser information indicating a riser height, whether the apparatus can move from one to another tread over the riser height. It is when the result of judgment is that the apparatus can move so that the apparatus can autonomously position itself to ascend or descend the stairs.

Also, the above robot apparatus may include a distance measuring means for acquiring the three-dimensional distance data, and thus it can autonomously climb stairs, if any detected.

Also, the above stair recognizing means may include a stair extracting means for detecting, based on the supplied plane information, a stair having a plane on

which the apparatus can move and outputting stair information not yet merged, and a stair merging means for statistically processing a plurality of yet-to-be-merged stair information supplied from the stair extracting means and which are temporally different from each other and outputting merged stair information. By these means, even a robot apparatus having a narrow field of view or not capable of recognizing stairs well with one time of stair extraction can recognize the stairs accurately in a wide range on the basis of the merged stair information resulted from the time-serial and statistic merging of such stair information.

Further, the above stair extracting means can recognize the dimensions and spatial positions of a tread on the basis of the plane information and output the resultant tread information as the yet-to-be-merged stair information, and the stair merging means can merge, when a group of two or more treads overlapping each other in a range larger than a predetermined threshold and whose difference in height from each other is smaller than a predetermined threshold has been detected from temporally continuous tread information, the tread information together into a single piece of tread information including tread information on all the treads in the tread group. Thus, a wide range of stair recognition is assured by merging the tread information to cover all the treads selected for merging of the information on them.

Furthermore, the stair recognizing means can recognize the dimensions and spatial position of a tread on the basis of the plane information, and take them as

the tread information. The tread information may include information on a front edge as the outermost edge, and information on a back edge as the innermost edge, of the tread in at least the moving direction. Since the stair recognizing means recognizes the front and back edges of the tread, it can recognize the tread accurately even in the case of a spiral stair, for example, to thereby enable the apparatus to climb the stairs.

Also, the tread information may include right- and left-margin information indicative of marginal regions adjacent to the right and left sides, respectively, of a safety zone defined by the front and back edges and in which it is estimated that it is highly probable that the apparatus can move, reference-point information indicative of the center of gravity of a region which is estimated, based on the plane information, to be a tread, three-dimensional coordinate information on a group of points forming together a plane which is to be a tread, etc. The stair climbing can be controlled accurately on the basis of these pieces of tread information.

Further, the stair recognizing means can extract the boundary of a plane on the basis of the plane information to calculate a polygon, and calculate the tread information from the polygon. For example, in case the field of view of the robot apparatus is narrow, when three-dimensional distance data is highly reliable, the polygon may be a convex polygonal region circumscribing the plane boundary extracted based on the plane information and thus may include an actually detected plane. On the other hand, if the distance data contains many noises, the polygon

may be a convex polygonal region inscribing the plane boundary extracted on the basis of the plane information and be included in the actually detected plane, so that the noise part can be cut off to accurately detect the tread.

Furthermore, the apparatus having moved to a predetermined position opposite to the back edge of a tread on which it is moving (will be referred to as "present tread" hereunder) can be controlled to climb the stairs. For example, in the case of stairs each having a tread whose front and back edges overlap each other, such as low-riser stairs, the apparatus can be controlled to climb the stairs by moving it toward the back edge. Similarly, the apparatus may be controlled to climb the stairs by moving toward the front edge.

Also, if the back edge of a present tread cannot be recognized, the stair-climb controlling means can control the apparatus to climb the stairs after the apparatus has moved to a predetermined position opposite to the front edge of a next tread to which the apparatus is to climb. For example, when the stair-climb controlling means detects a stair when the apparatus is moving on a floor surface, the back edge of the floor surface does not overlap the front edge of a first tread of the stair as the case may be. In such a case, the stair-climb controlling means can control the apparatus to climb the stair by moving it toward the front edge of the next tread to which the apparatus is to climb.

Also, when the stair-climb controlling means detects a next tread to which the apparatus is to move, it can control the apparatus to climb the stairs by



performing a series of operations for moving to a predetermined position opposite to the tread to which the apparatus is to move. Each time the apparatus moves to a new tread, the stair-climb controlling means can control the apparatus to climb the tread by making a search for the tread, align itself to the tread and approach to the tread (search-align-approach operation).

Further, if the stair-climb controlling means cannot extract a next tread and subsequent treads with the apparatus standing in a present position or on a present tread, it can find, based on already acquired stair information, a next tread to which the apparatus is to move. By acquiring stair information on a several-steps higher or lower tread in advance, the stair-climb controlling means can control the apparatus to climb the stairs even if the apparatus is not constructed to acquire information on its nearest surroundings and namely its field of view is narrow.

Also, after the apparatus has moved to a predetermined position opposite to the back edge of a present tread, the stair-climb controlling means can detect a next tread to which the apparatus is to move, have the apparatus move to a predetermined position opposite to the front edge of the tread and have the apparatus climb to the tread. Thus, based on the data on the front and back edges, the stair-climb controlling means can control the apparatus to climb even a spiral stair in which both the edges are not parallel to each other.

Further, the stair-climb controlling means can control the stair climbing of the apparatus on the basis of parameters defining a position of the locomotion

means in relation to a tread. The parameters can be determined on the basis of a height of foot lifting or lowering, for example, of the leg. The stair-climb controlling means may include a parameter selecting means for selecting parameter values for an intended stair-ascending or -descending motion. Thus, the stair-climb controlling means can control both the stair-ascending and -descending motions in the same manner just by changing the values of parameters for stair-ascending or -descending motion, whichever is intended.

Also, the plane segmenting/extracting means includes a line fitting means for fitting line segments (will be referred to as "line" hereunder wherever appropriate) for each group of distance data points estimated to be in one plane in a three-dimensional space, and a planar region growing means for extracting a plurality of lines estimated to be in one plane from the group of lines fitted by the line fitting means and calculating a plane from the extracted plurality of lines.

The line fitting means can fit lines adaptively to the distribution of the distance data points. It extracts lines based on the fact that three-dimensional distance data in one plane are in one straight line. Since the distribution of distance data points is varied under the influence of noises or the like at this time, the line fitting means fits lines adaptively to the distribution of distance data (adaptive line fitting), to thereby enable an accurate line fitting robustly against the noises and determine a plane from the fitted plurality of lines by the scan line grouping method. Thus, a plane can accurately be extracted without taking a



plurality of normally existing planes as a single plane or taking a normally existing plane as a plurality of planes, which would otherwise be caused by the influence of the noises or the like.

Further, the line fitting means can extract a group of distance data points estimated to be in one plane on the basis of the distance between the distance data points, and re-estimate, based on the distribution of the distance data points in the distance data point group, whether the distance data point group exists in one plane. It can accurately fit lines by extracting a distance data point group once on the basis of a distance between the distance data points in the three-dimensional space and estimating, based on the data point distribution, whether the distance data point group lies in one plane.

Furthermore, the line fitting means can extract a first line from the group of distance data points estimated to lie in one plane, take a distance data point in the group, whose distance from the first line is longest, as a point of interest, extract a second line from the distance data point group in case the distance is smaller than a predetermined threshold, judge whether a larger number of distance data points than a predetermined number exist continuously at one side of the second line, and segment the distance data point group by the point of interest in case the larger number of distance data points than the predetermined number exist continuously. For example, the line fitting means takes a line connecting end points of an extracted data point group as the first line, and generates a second line by the

least-squares method, for example, in case there exists a point whose distance from the first line is long. In case there exists a plurality of data points continuously at one side of the second line, the line fitting means can assume that the data point group is in the zig-zag shape, for example, in relation to the line and thus determine that the extracted data point group is biased, and segment the data point group by the point of interest or the like.

Also, the planar region grouping means can select more than one line estimated to be in one plane and calculate a reference plane, search lines estimated to be in the plane in which the reference plane lies as grouping lines from the group of lines, update the reference plane with the grouping lines and repeat the grouping of the region of the reference plane, and output the updated plane as an updated one. Thus, the planar region grouping means can make the planar region growing and plane updating with the lines estimated to lie in one plane.

Further, the planar region grouping means may includes a plane recalculating means for recalculating a plane from the group of distance data points except for ones whose distance from the updated plane is larger than a predetermined threshold, if any, existing in the distance data point group in the updated plane. Since the updated plane is taken as a plane in which all lines are uniformly distributed, it is possible to have a result of extraction less influenced by noises or the like by recalculating a plane from the group of distance data points except for ones largely different from the plane in which all the lines are uniformly

distributed.

Furthermore, the planar region grouping means can estimate, based on a difference between a lines-determined plane and reference plane, whether the lines lie coplanar with the reference plane. Namely, it is possible to extract a plane more accurately by judging, based on a root mean square residual of the plane equation, for example, whether the extraction has been influenced by noises or the lines are not coplanar with the reference plane.

Also the above object can be attained by providing a method of controlling the motion of a robot apparatus enabled by a locomotive means to move, the method including, according to the present invention, the steps of:

detecting a plane or a plurality of planes included in an environment surrounding the apparatus from three-dimensional distance data and outputting the detected plane or planes as plane information;

recognizing, based on the plane information, a stair having a plane on which the apparatus can move and outputting stair information including information on the tread and riser of the stair; and

controlling the stair climbing of the apparatus by judging, based on the stair information, whether the apparatus can climb the stair and having the apparatus autonomously position itself in relation to the stair tread when the stair is determined to be climbable.

Also the above object can be attained by providing a locomotion mechanism

movable by locomotion means, the mechanism including according to the present invention:

a plane segmenting/extracting means for detecting a plane or a plurality of planes included in an environment surrounding the apparatus from three-dimensional distance data and outputting the detected plane or planes as plane information;

a stair recognizing means for recognizing, based on the plane information, a stair having a plane on which the apparatus can move and outputting stair information including information on the tread and riser of the stair; and

a stair-climb controlling means for controlling the stair climbing of the apparatus by judging, based on the stair information, whether the apparatus can climb the stair and having the apparatus autonomously position itself in relation to the stair tread when the stair is determined to be climbable.

In the robot apparatus including legs and the like as locomotion means and locomotion mechanism used in the robot apparatus as above, it is possible to judge, based on tread information on the dimensions and position of a stair tread, whether the foot base of the robot apparatus has such a size that the foot base can be placed on the tread, judge, based on riser information indicative of a step height, whether the robot apparatus can move to the tread of such a height, and climb the stairs by autonomously positioning the robot apparatus in case it has been determined that the robot apparatus can move to the tread.

These objects and other objects, features and advantages of the present invention will become more apparent from the following detailed description of the preferred embodiments of the present invention when taken in conjunction with the accompanying drawings.

#### Brief Description of the Drawings

FIG. 1 explains an example of the conventional stair-climb controlling techniques;

FIG. 2 explains another example of the conventional stair-climb controlling techniques;

FIGS. 3A and 3B explain still another example of the conventional stair-climb controlling techniques;

FIG. 4 is a perspective view of a robot apparatus according to the present invention;

FIG. 5 schematically illustrates the degrees of freedom of the joints of the robot apparatus;

FIG. 6 schematically illustrates the control system employed in the robot apparatus;

FIG. 7 is a functional block diagram of a system in which the robot apparatus makes operations from acquisition of stereo data to stair climbing;

FIG. 8A schematically illustrates the robot apparatus taking a picture of its

surroundings and FIG. 8B shows the dimensions of the foot bases of the robot apparatus;

FIG. 9 explains the stair extraction, FIG. 9A shows the front view of the stair, FIG. 9B shows the side elevation of the stair, and FIG. 9C shows the perspective view of the stair;

FIG. 10 explains another embodiment of the stair extraction, FIG. 10A shows the front view of the stair, FIG. 10B shows the side elevation of the stair, and FIG. 10C shows the perspective view of the stair;

FIG. 11 shows examples of the results of the extraction of the stairs in FIG. 9, FIG. 11A schematically illustrates the segmentation of the image of the stair in FIG. 9, and FIGS. 11B to 11D show the three-dimensional distance data acquired from images in FIG. 11A;

FIG. 12 shows other examples of the results of the extraction of the stairs in FIG. 10, FIG. 12A schematically illustrates the segmentation of the image of the stair in FIG. 10, and FIGS. 12B to 12D show the three-dimensional distance data acquired from images in FIG. 12A;

FIG. 13A schematically illustrates the segmentation of the image of the stair, FIG. 13B shows the result of the extraction of the four plane regions A to D acquired from the three-dimensional distance data of FIG. 13A, showing another example of the result of the extraction of the stair;

FIG. 14 shows functional blocks of the stair recognition unit;

FIG. 15 shows a flow of operations made in the stair extraction;

FIG. 16 schematically illustrates a convex polygon;

FIG. 17 schematically illustrates the *Melkman's* algorithm;

FIG. 18 is a schematic illustration for explaining how to determine a polygon by the *Sklansky's* algorithm;

FIG. 19 is a schematic illustration for explaining a problem taking place with a non-polygonal stair, in which FIG. 19A shows an input plane and FIG. 19B shows the result of representation of the non-polygonal stair in a polygonal form by a convex hull.

FIG. 20 is a schematic illustration for explaining a method of determining a polygon including an input plane by smoothing, in which FIG. 20A shows the input plane, FIG. 20B shows a polygon smoothed by removing discontinuous gaps from a polygon representing the input plane and FIG. 20C shows a polygon further smoothed by fitting line segments of the polygon shown in FIG. 20B.

FIG. 21 shows an example of the program for determining a polygon including an input plane by smoothing with the close-gaps and fit-line-segments operations.

FIGS. 22A and 22B show schematic illustration for explaining how to calculate stair parameters;

FIG. 23 is a schematic illustration for explaining a finally recognized tread and stair parameters;



FIGS. 24A and 24B schematically illustrate stairs;

FIG. 25 shows a flow of operations made in stair-information merging;

FIG. 26 is a schematic illustration for explaining the merging of stair data overlapping each other;

FIG. 27 explains an align mode of operation;

FIG. 28 is a schematic illustration for explaining an approach mode of operation;

FIG. 29 shows a flow of operations made in a stair climb mode of operation;

FIG. 30 shows a flow of operations made in the search, align and approach modes of operation (search-align-approach);

FIG. 31 shows a flow of operations made in the stair climb mode;

FIG. 32 schematically illustrates a tread the robot apparatus has recognized or is going to recognize;

FIG. 33 shows a flow of operations made in the stair climb mode;

FIG. 34 also shows a flow of operations made in the stair climb mode;

FIG. 35A explains the relation between a tread the robot apparatus has recognized and the foot base and FIG. 35B shows the measurement of each unit;

FIG. 36 are traced drawings of the robot apparatus ascending stairs;

FIG. 37 are traced drawings of the robot apparatus descending the stairs;

FIG. 38 shows a geometric relation between a single step and the foot bases of the robot apparatus;



FIG. 39 shows a geometric relation between a single concavity and the foot bases of the robot apparatus;

FIG. 40 shows functional blocks of the plane segmentation/extractor unit in this variant;

FIG. 41 explains a robot apparatus including a means for imparting a texture;

FIG. 42 explains how to extract planes by the scan line grouping method in a variant of the aforementioned embodiment;

FIG. 43 shows a flow of operations made in the plane segmentation/extraction by the scan line grouping method;

FIG. 44 shows a flow of operations made in the plane segmentation/extractor unit in the variant;

FIG. 45 shows how distance data points are distributed, in which FIG. 45A shows the data being distributed in the zig-zag shape in relation to a line and FIG. 45B shows the data being distributed evenly near a line under the influence of noises or the like;

FIG. 46 shows a flow of operations made in the zig-zag shape discrimination method in the variant;

FIG. 47 shows an exemplary program of the zig-zag shape discrimination;

FIG. 48 is a block diagram of the zig-zag shape discriminator;

FIG. 49 is a schematic illustration for explaining the region growing in the variant;

FIG. 50 shows a flow of operations made in seed-region search and region growing in the region grouping unit in the variant;

FIG. 51 schematically illustrates examples equal in distance between an end point and straight line to each other but different in root mean square residual *rms* of the plane equation from each other, in which FIG. 51A shows an example in which the line is off the plane under the influence of noises or the like and FIG. 51B shows an example in which there exists another plane in which the line is;

FIG. 52 shows selection of a seed region;

FIG. 53 shows the region growing;

FIG. 54A is a schematic illustration of a floor surface viewed from the robot apparatus being in upright position, FIG. 54B shows three-dimensional distance data with the vertical axis taken as  $\underline{x}$ , horizontal axis as  $\underline{y}$  and the density of each data point taken as  $\underline{z}$ , and straight lines extracted from a group of data points lying in one plane or line-directional pixel train by the line fitting process, and FIG. 54C shows a planar region determined by the region growing from the group of straight lines shown in FIG. 54B;

FIG. 55 explains the difference in result of performance between the plane segmentation/extraction method according to the variant and conventional plane segmentation/extraction method when a step is placed on a flat surface, in which FIG. 55A schematically illustrates an image being viewed, FIG. 55B shows experimental conditions, FIG. 55C shows the results of plane

segmentation/extraction by the plane segmentation/extraction method according to the variant, and FIG. 55D shows the results of plane segmentation/extraction by the conventional plane segmentation/extraction method; and

FIG. 56A schematically illustrates an image of a floor surface, and FIGS. 56B and 56C show lines extracted by the line fitting method according to the variant and by the conventional line fitting method, respectively, from the horizontal and vertical distance data point trains, respectively, included in the three-dimensional distance data acquired by taking a picture of the floor surface shown in FIG. 56A.

#### Best Mode for Carrying out the Invention

The embodiment which will be illustrated and explained hereunder is an application of the present invention to an autonomous locomotion robot apparatus including a step recognition unit to recognize a step such as a stair or the like existing in the surrounding environment.

The robot apparatus according to this embodiment of the present invention is capable of recognizing stairs on the basis of a plurality of planes extracted from distance data acquired by a stereo vision system or the like and climbing the stairs on the basis of the result of stair recognition.

#### (1) Robot apparatus

First, a bipedal walking robot apparatus will be illustrated and explained as

an example of the robot apparatus of this type. The robot apparatus is a practical-use robot which supports human activities in various living environments and other daily lives. It is also an entertainment robot capable of acting correspondingly to its internal status (anger, sadness, joy, pleasure, etc.) and imitating basic behaviors of the human being. It should be noted that although the present invention will be explained as to a bipedal walking robot apparatus as an example, a stair recognition unit used in the robot apparatus according to the present invention, if installed in a robot apparatus of any other legged locomotion type than the bipedal type, can have the legged locomotion robot apparatus climb stairs.

FIG. 4 is a perspective view of a robot apparatus according to the embodiment of the present invention, showing the concept of the robot apparatus. As shown in FIG. 4, the robot apparatus, generally indicated with a reference numeral 201, includes a body unit 202 and a head unit 203, two arm units 204R and 204L, and two leg units 205R and 205L, each connected in a predetermined position to the body unit 202 (note that "R" and "L" indicate "right" and "left", respectively; this is also true for the following illustration and description).

The degrees of freedom of the joints of the robot apparatus 201 are schematically illustrated in FIG. 5. The neck joint supporting the head unit 203 has three degrees of freedom including a neck joint yaw axis 101, neck joint pitch axis 102 and neck joint roll axis 103.

Also, each of the arm units 204R and 204L forming together each upper limb includes a shoulder joint pitch axis 107, shoulder joint roll axis 108, upper arm yaw axis 109, elbow joint pitch axis 110, lower arm yaw axis 111, wrist joint pitch axis 112, wrist joint roll axis 113 and a hand 114. The hand 114 is actually a multi-joint, multi-freedom structure including a plurality of fingers. However, since the motion of the hand 114 little contributes to or influences the control over the attitude and walking of the robot apparatus 201, it is assumed herein for the simplicity of the explanation that hand 114 has no degree of freedom. Therefore, each arm unit has seven degrees of freedom.

Also, the body unit 202 has three degrees of freedom including a body pitch axis 104, body roll axis 105 and body yaw axis 106.

Also, each of the leg units 205R and 205L forming together each lower limb includes a hip joint yaw axis 115, hip joint pitch axis 116, hip joint roll axis 117, knee joint pitch axis 118, ankle joint pitch axis 119, ankle joint roll axis 120 and a foot base 121. It is defined herein that the intersection of the hip joint pitch axis 116 and hip joint roll axis 117 is the hip joint position of the robot apparatus 201. The foot base or sole 121 of the human body is actually a multi-joint, multi-freedom structure. It is assumed herein for the simplicity of explanation that the foot base of the robot apparatus 201 has no degree of freedom. Therefore, each leg unit has six degrees of freedom.

As above, the robot apparatus 201 has a total of 32 degrees of freedom (= 3

+  $7 \times 2 + 3 + 6 \times 2$ ). However, the robot apparatus 201 destined for entertainment has not always the 32 degrees of freedom. It is of course that the entertainment robot apparatus can be designed to have an appropriately smaller or larger number of degrees of freedom, namely, of joints, depending upon the constraints on the design and manufacture, specifications required for the intended robot apparatus, etc.

Each of the above-mentioned degrees of freedom of the robot apparatus 201 is actually implemented by an actuator. To provide a robot apparatus having a shape approximate to the natural shape of the human body without excessive apparent bulge and control the attitude of an unstable structure such as a bipedal walking robot apparatus, the actuator should preferably be compact and lightweight.

The above-mentioned robot apparatus has provided in the body unit 202, for example, a control system to control the motions of the entire robot apparatus itself. FIG. 6 schematically illustrates the control system employed in the robot apparatus 201. As shown in FIG. 6, the control system includes a thought control module 200 dynamically responsive to a user's input or the like to control the emotion judgment and sentiment expression, and a motion control module 300 to control the whole-body coordinated motion of the robot apparatus 201, such as the driving of actuators 350.

The thought control module 200 is an independent driving type information

processing unit including a central processing unit (CPU) 211 to make calculations as to the emotion judgment and sentiment expression, random-access memory (RAM) 212, read-only memory (ROM) 213, external storage unit (such as a hard disk drive) 214, etc. and which can make a self-contained processing within itself.

The thought control module 200 determines a current sentiment and will of the robot apparatus 201 according to an external stimulus or the like such as image data supplied from an image input unit 251, sound data supplied from a sound input unit 252, etc. That is, the thought control module 200 can recognize the facial expression of the user of the robot apparatus 201 on the basis of the input image data and make an action corresponding to the user's facial expression by reflecting the information on the sentiment and will of the robot apparatus 201. The image input unit 251 included in the present invention includes a plurality of CCD (charge-coupled device) cameras, for example, and can acquire a range image based on images captured by the cameras. Also, the sound input unit 252 includes a plurality of microphones, for example.

The thought control module 200 issues a command to the motion control module 300 for execution of a sequence of motions or actions based on the determined will, that is, motions of the four limbs.

The motion control module 300 is an independent drive type information processor including a CPU 311 to control the whole-body coordinated motion of the robot apparatus 201, RAM 312, ROM 313, external storage unit (such as hard



disk drive) 314, etc. and which can make a self-contained processing within itself. Also, the external storage unit 314 can store a walking pattern, target ZMP trajectory and other action plan calculated on offline basis, for example.

The motion control module 300 has connected thereto, by a bus interface (I/F) 310, various units including actuators 350 to implement degrees of freedom of the joints located on the whole body of the robot apparatus 201 as shown in FIG. 5, distance measuring sensors (not shown) to measure a distance from the robot apparatus 201 to an object, attitude sensor 351 to measure the attitude and inclination of the body unit 202 of the robot apparatus 201, landing checking sensors 352 and 353 to detect when the robot apparatus 201 lifts the right or left foot base 121 from a surface or places the foot base on the surface, load sensor provided on the foot base 121, power controller 354 to control the power source such as a battery or the like, etc. It should be noted here that the attitude sensor 351 is for example a combination of an acceleration sensor and gyro sensor and each of the landing checking sensors 352 and 353 is formed from a proximity sensor, microswitch or the like.

The thought control module 200 and motion control module 300 are built on a common platform and connected to each other by a bus interface 210 and the bus interface 310.

The motion control module 300 controls the whole-body coordinated motion made by the actuators 350 to implement an action commanded from the thought



control module 200. More specifically, the CPU 311 calls a motion pattern corresponding to a motion commanded from the thought control module 200 from the external storage unit 314 or internally creates a motion pattern. Then, the CPU 311 sets, according to the specified motion pattern, a foot motion, ZMP (zero moment point) trajectory, body motion, upper-limb motion, horizontal position and height of the hip, etc. and transfers command values commanding motions corresponding to the settings to the actuators 350.

Also, the CPU 311 can adaptively control the whole-body coordinated motion of the robot apparatus 201 by detecting the attitude and inclination of the body unit 202 of the robot apparatus 201 on the basis of an output signal from the attitude sensor 351, while detecting, based on output signals from the landing checking sensors 352 and 353, whether the leg unit 205R or 205L is in swinging phase or in supporting phase. Further, the CPU 311 controls the attitude and motion of the robot apparatus 201 for the ZMP (zero moment position) to always be positioned toward the center of ZMP-stable region.

Also, the motion control module 300 is designed to return to what extent an action corresponding to a will determined in the thought control module 200 has been implemented, that is, the progress of operation, to the thought control module 200. Thus, the robot apparatus 201 can judge its internal and surrounding conditions on the basis of a control program, and act autonomously.

## (2) Motion controlling method for the robot apparatus

In the aforementioned robot apparatus, the head unit 203 can have installed therein a stereo vision system to acquire three-dimensional distance information on the surrounding environment. Next, there will be explained a series of operations made using the three-dimensional distance data acquired by the stereo vision system from the surrounding environment to extract planes recognizing stairs on the basis of the result of plane segmentation/extraction and controlling the robot apparatus to climb the stairs on the basis of the result of stair recognition.

FIG. 7 is a functional block diagram of a system in which the robot apparatus makes a process from the extraction of stereo data down to stair climbing. As shown in FIG. 7, the robot apparatus includes a stereo vision system 1 to acquire three-dimensional distance data, a plane segmentation/extractor unit 2 which is supplied with stereo data D1 from the stereo vision system 1 and detects intra-environment planes from the stereo data D1, a stair recognition unit 3 to recognize stairs from plane data D2 supplied from the plane segmentation/extractor unit 2, and a stair climber 4 to output a motion control command D5 for causing the robot apparatus to climb the stairs on the basis of stair data D4 which is the result of recognition supplied from the stair recognition unit 3.

The robot apparatus views the surrounding environment by means of the stereo vision system 1 and outputs, as an image, the stereo data D1 which is three-dimensional distance information calculated from the parallax between both the eyes. More particularly, the robot apparatus compares pixels included in

image inputs from two cameras, right and left, equivalent to the human eyes, estimates the distance to an object on the basis of the parallax between the cameras, and outputs three-dimensional distance information as an image (range image). The robot apparatus can recognize a plurality of planes existing in the environment by segmenting the range image and extracting the planes by means of the plane segmentation/extractor unit 2. Further, the plane recognition unit 3 extracts, from these planes, a plane the robot apparatus can climb, recognizes a stair on the basis of the plane, and supplies the stair data D4 to the stair climber 4. The latter will use the stair data D4 to output the motion control command D5 which implements the stair climbing.

FIG. 8A is a schematic illustration showing the robot apparatus 201 taking a picture of the surrounding environment. On the assumption that the floor-surface plane is  $x$ - $y$  and the height direction is  $z$ , the robot apparatus 201 having an image input unit (stereo camera) installed in the head unit 203 thereof can view a predetermined field in front thereof as shown in FIG. 8A.

In the robot apparatus 201, the CPU 211 is supplied with a color image and parallax image from the image input unit 251 and all sensor data such as joint angles and the like from the actuators 350 to execute software for variety of operations.

The software executed in the robot apparatus 201 according to this embodiment is configured in units of an object to permit various operations for

recognition of the position of the robot apparatus itself, moving distance of the robot apparatus, obstacle or obstacles lying around the robot apparatus, environmental map, etc., and outputting of a train of actions to finally be done by the robot apparatus. It should be noted that this embodiment of the present invention uses two coordinates including a camera coordinate system of a world-standard type whose origin is a position predetermined based on a specific object such as a landmark (will also be referred to as "absolute coordinate" hereunder) and a robot-center coordinate system whose center (coordinate origin) is the robot apparatus itself (will also be referred to as "relative coordinate" hereunder).

In the stereo vision system 1, joint angles determined from the sensor data are utilized to transform the robot-center coordinate system having the robot apparatus 201 fixed at the center thereof into the coordinate system of the image input unit 251 provided in the head unit 203 at times when the image data including the color image, parallax image from the stereo camera, etc. have been captured. In this case, the stereo vision system 1 in this embodiment derives a homogeneous transformation matrix etc. of the camera coordinate system from the robot-center coordinate system and outputs a range image formed from homogeneous transformation matrix and corresponding three-dimensional distance data.

The robot apparatus according to this embodiment can recognize stairs included in its field of view, and climb the stairs on the basis of the result of

recognition (will be referred to as "stair data" hereunder). Therefore, for climbing stairs, the robot apparatus has to make judgment as to the dimensions of the stairs to see whether the stair dimensions are larger than the foot bases of the robot apparatus itself, whether the step height is such that the robot can climb, and the like.

The robot apparatus according to this embodiment has the foot bases whose dimensions are as shown in FIG. 8B. In FIG. 8B, on the assumption that the forward-moving direction of the robot apparatus 201 is an x-axial direction and a direction parallel to the floor surface and perpendicular to the x-axial direction is a y-axial direction, the y-directional width of both the feet when the robot apparatus 201 stands upright is taken as feet base width, the width of a part before the ankle (connection between the leg and foot base) and having the dimensions of the foot base is taken as "*foot\_front\_size*" (front-part size of the foot) and the width of a part behind the ankle and having is taken as "*foot\_back\_size*" (back-part size of the foot).

The robot apparatus 201 extracts stairs as shown in FIGS. 9 and 10 in the environment around itself. FIGS. 9A and 10A are front views of the stairs, FIGS. 9B and 10B are side elevations of the stairs, and FIGS. 9C and 10C are perspective views of the stairs.

Note here that a plane or surface on which the human being, robot apparatus or the like steps when climbing stairs (on which the feet or moving legs are placed)

is called "tread" and the height from one tread to a next tread (one step height) is called "riser". Also it should be noted that in the following, the stairs will be counted as one step, second step, ... as the robot ascends the stairs from the level near the ground surface.

In stairs ST1 in FIG. 9, three steps are included, the riser is 4 cm high, the first and second treads are 30 cm wide and 10 cm deep, and only the topmost (third) tread is 30 cm wide and 21 cm deep. Also in stairs ST2 shown in FIG. 10, three steps are included, the riser is 3 cm high, the first and second treads are 33 cm wide and 12 cm deep, and only the topmost (third tread) is 33 cm wide and 32 cm deep. The robot apparatus recognizes these stairs as will be described later.

The plane segmentation/extractor unit 2 detects the distance information (stereo data D1) supplied from a distance measuring unit (range finder) such as a stereo vision system and extracts a plurality of planes existing in the environment, and outputs the plane data D2. The plane segmentation/extraction can be implemented by the scan line grouping method which will be described later as well as by the conventional plane segmentation/extraction technique using the Hough transformation. To extract a plurality of planes such as stairs from distance data including noises however, a scan line grouping method which will be described later or the like should preferably be adopted since the method will assure an accurate extraction of planes.

FIGS. 11 and 12 show examples of the result of stair extraction. The



examples shown in FIGS. 11 and 12 are results of stair extraction made by acquiring three-dimensional distance data from images of the stairs shown in FIGS. 9 and 10, and segmenting the data to extract planes. More specifically, FIG. 11A schematically illustrates the segmentation of the image of the stair in FIG. 9, and FIGS. 11B to 11D show the three-dimensional distance data acquired from images in FIG. 11A. Also, FIG. 12A schematically illustrates the segmentation of the image of the stair in FIG. 10, and FIGS. 12B to 12D show the three-dimensional distance data acquired from images in FIG. 12A. As in the examples shown in FIGS. 11 and 12, all the treads could be extracted as planes. FIG. 11B shows an example in which the first, second and third treads have been extracted through the three-dimensional distance data segmentation/plane extraction process. As will be seen in FIG. 12B, a part of the floor surface has also been extracted successfully as a plane.

More specifically, as shown in FIG. 13A when the planes are extracted through the segmentation of a range image of the stairs ST2 and extraction, for example, regions A to D are extracted as planes indicative of the floor surface, first step, second step and third step, respectively, as shown in FIG. 13B. A group of points included in the same region included in the regions A to D indicates a group of distance data points estimated to form the same plane.

The stair recognition unit 3 is supplied with the plane data D2 extracted by the plane segmentation/extractor unit 2 and recognizes the shape of the stairs, that

is, the tread dimensions, step height (riser height), etc. The stair recognition unit 3 recognizes, as stair data, an edge (front edge) (will be referred to as "FE" hereunder) at the near side (near the robot apparatus) of a region (polygon) included in the tread recognized by the robot apparatus 201 and an edge (back edge) (will be referred to as "BE" hereunder) at the far side (far from the robot apparatus) of the tread. Then, the stair climber 4 uses the stair data thus acquired to control the stair climbing of the robot apparatus.

Next, the control of stair climbing of the robot apparatus will be described in detail. It should be noted that in the following, firstly the stair recognition by the robot apparatus will be described, secondly the plane segmentation/extraction by the robot apparatus will be described concerning the recognized stairs, and finally the plain recognition by the scan line grouping method will be described as a concrete method of plane segmentation/extraction.

FIG. 14 is a block diagram of the stair recognition unit shown in FIG. 7. As shown in FIG. 14, the stair recognition unit 3 includes a stair extraction unit 5 to extract stairs from the plane data D2 supplied from the plane segmentation/extractor unit 2, and a stair merging unit 6 to recognize more accurately the stairs by merging timer-series data of the stair data D3 extracted by the stair extraction unit 5, namely, a plurality of stair data D3 extracted at different times. The stair data D4 resulted from the merging by the stair merging unit 6 is delivered at the stair extraction unit 5.



The stair extraction unit 5 extracts stairs from the plane data D2 supplied from the plane segmentation/extractor unit 2. The plane data D2 supplied from the plane segmentation/extractor unit 2 includes a plurality of information per plane as will be shown below. Namely, the stair extraction unit 5 will be supplied with plane data per a plurality of planes extracted from images acquired by the stereo vision system 1.

That is, the plane data D2 includes the following information per plane:

- 1-1 Number of supporting points
- 1-2 Central point of a plane
- 1-3 Plane parameters (normal vector and distance from the origin)
- 1-4 Edges of polygon, defining a plane

The robot apparatus selects a plane such as a floor surface or tread on which it is placing its foot or feet and generally flat surfaces and calculates the following information (will be referred to as "stair parameters" hereunder):

- 2-1 Front edge (FE) and back edge (BE)
- 2-2 Step height

The front edge FE and back edge BE recognized by the robot apparatus indicate the edges (lines) of a stair tread. On the assumption that the robot apparatus is facing a polygon, an edge, near the robot apparatus (near-side edge), of the polygon is taken as the front edge FE, and an edge, far from the robot apparatus (far-side edge), of the polygon is taken as the back edge BE. That is, a smallest

polygon including all points defining a plane, for example, may be determined, and a near edge of the polygon may be taken as the front edge FE while a far edge may be taken as the back edge BE. Information on the front and back edges FE and BE may be taken as information on these end points. Also, information such as stair width W, stair length L, etc. can be acquired from the polygon. Further, a stair height (riser) may be taken a difference in height between centers of two planes in the supplied plane data D2 or a difference in height between two centers of gravity with which the polygon has been determined. It should be noted that the riser may be a difference in height between the back edge BE of a former tread and the front edge FE of a latter tread.

Also, in the embodiment, the front and back edges FE and BE, and regions adjacent to the right and left of a region (safety zone) between the front and back edges FE and BE and where it is estimated that the robot apparatus is highly likely to move are recognized as margins. These margins are determined as will be described later. By determining the margins, it is possible to widely recognize the tread region where it is estimated that the robot apparatus can move. Further, information such as the number of data points in a group defining a tread, information on a reference point defining one centroidal point, etc. may be taken as the stair data D3.

Planes (stairs) which meet the following requirements are extracted from the above stair data:

3-1 Front and back edges FE and BE are larger in length than a predetermined threshold

3-2 Stair height is smaller than a predetermined threshold

It is preferable to extract planes which additionally meet the following requirements:

3-3 Stair width W is larger than a predetermined threshold

3-4 Stair length L is larger than a predetermined threshold

FIG. 15 shows a flow of operations made in the stair extraction by the stair extraction unit 5. As shown in FIG. 15, the stair climber 4 first judges, based on the plane parameters in the supplied plane data, whether the input plane which a supporting surface, for example, is horizontal or whether it is a plane on which the robot apparatus can walk or move (in step S1). Conditions for the plane to be horizontal or a plane on which the robot apparatus can move should be set according to the functions the robot apparatus is designed to have. For example, on the assumption that the plane vector of an input plane is  $n(n_x, n_y, n_z)$ , the plane can be judged to be horizontal when  $|\sin^{-1}n_z| > \min_{th}$  (where  $\min_{th}$  is a threshold for judgment of the plane to be horizontal or not). For example, when  $\min_{th} = 80$  deg. with distance data used, accuracy of plane segmentation/extraction, etc. taken in account, a plane having a maximum angle of  $\pm 10$  deg. in relation to the horizontal plane may be judged to be horizontal. Also, if an input plane has an angle of about  $\pm 30$  deg. in relation to the horizontal plane but is a plane on which the robot

apparatus can walk or move, planes whose angles are within these ranges ( $\pm 10$  or  $\pm 30$  deg) should be extracted. If the stair climber 4 has determined the input plane not to be horizontal (No in step S1), it exists this operation with informing the failure in plane segmentation/extraction, and repeats the operation with next plane data.

If the stair climber 4 has determined the input plane to be horizontal (Yes in step S1), it will recognize the edges (shape) of the plane. The stair climber 4 determines, in step S2, a polygon including an input plane by the convex hull algorithm or smoothing by noise removal as in the well-known technique such as the *Sklansky's* algorithm (J. Sklansky, "Measuring Concavity on a Rectangular Mosaic", IEEE Trans. Comput. 21, 1974, pp. 1355-1364), *Melkman's* algorithm (Melkman A., "On-Line Construction of the Convex Hull of a Simple Polygon", Information Processing Letters 25, 1987, p. 11), for example. Then, the stair climber 4 determines front and back border lines of this polygon to be stair parameters such as a front edge and back edge in step S3. Further in this embodiment, the stair climber 4 determines, based on the front and back edges, a width W and length L of a plane defined by a stair tread as the border lines and then judges whether the values W and L are larger than predetermined thresholds, in step S4. If the stair climber 4 has determined the tread width and length not to be larger than the thresholds (No in step S4), it will determine the plane not to be any plane on which the robot apparatus can move and repeat the operations in steps S1

to S4 with next plane data.

If the stair climber 4 has determined the plane width and length to be larger than the thresholds (No in step S4), it will determine the plane to be a plane on which the robot apparatus can move, calculate right and left margins in step S5 and outputs these information as the stair data D3.

Next, there will be explained how to form a polygon including an input plane as a convex hull algorithm as in step S2. FIG. 16 schematically illustrates a convex polygon. FIG. 16A shows a region including all supporting points determined to be in one input plane (group of distance data points regarded as included in continuous regions in one plane), and FIG. 16B shows a convex polygon determined from the graphic shown in FIG. 16A. The convex polygon shown in FIG. 16B can be regarded as using a convex hull to determine a minimum convex set including given plane graphics (regions including supporting points). It should be noted that a point indicated with G, which will be described in detail later, is used to determine the width W of a tread. For example, it is a point (reference point) such as a center of gravity of a region including supporting points.

Typical examples of the algorithm for determining a convex polygon by such a convex hull include the aforementioned *Melkman's* algorithm, *Sklansky's* algorithm, etc. FIG. 17 is a schematic illustration for explanation of the *Melkman's* algorithm. As shown in FIG. 17, three of points, P1, P2 and P3, included in a given graphic are extracted, the points P1 and P2 are connected by a

line to each other, and there are drawn straight lines passing by the points P1 and P3 and the points P2 and P3, respectively. Thus, the graphic is zoned by these lines into five areas AR1 to AR5 including a triangle AR4 formed by the three points P1, P2 and P3. Then, a polygon is reformed by judging in which area the selected point P4 is included. This operation is repeated to update the convex polygon. For example, if the point P4 exists in the area AR1, an area defined by a line passed through the points P1, P2, P4 and P3 in this order will be a new convex polygon. Also, if the point P4 exists in the areas AR3 and AR4, the convex polygon is updated to an area defined by a line passed through the points P1, P2, P3 and P4 in this order and an area defined by a line passed through the points P1, P4, P2 and P3 in this order, respectively. On the other hand, if the point P4 is included in the area AR4, that is, inside the convex polygon, the latter will not be updated. Also, if the point P4 exists in the area AR2, the convex polygon is updated to an area defined by a line passed through the points P1, P2 and P3 in this order except for the point P3. In this embodiment, a convex polygon may be formed for all the supporting points with an area included in each point being taken in consideration.

FIG. 18 is a schematic illustration for explaining how to determine a polygon by the *Sklansky's* algorithm. A polygon extracted using the *Sklansky's* algorithm is called "Weakly externally visible polygon" which needs less computational complexity than the *Melkman's* algorithm and can thus be calculated at a higher



velocity.

In case a convex polygon including a given graphic is determined, a half straight line is drawn from an arbitrary point  $x$  on the boundary of the graphic 131 to a circle 132 including a graphic 141 as shown in FIG. 18A. At this time, in case a half straight line not crossing the graphic 131, of the half straight line drawn from the point  $x$  to the circle 132, can be drawn, the point should be a point included in the boundary of the convex polygon. On the other hand, in case a half straight line is drawn from another arbitrary point  $y$  on the boundary of a given graphic 133 to a circle 134 including the graphic 133 as shown in FIG. 18B, a half straight line not crossing the graphic 133 cannot be drawn. In this case, the other point  $y$  should not be included in the boundary of the convex polygon. When a graphic is determined only from the points selected by judging the points one after another to see whether each of them forms the convex polygon as above, a graphic as shown in FIG. 16A will be provided.

By determining a convex polygon including the graphic, it is possible to determine a convex polygon as shown in FIG. 16B can be obtained. According to this embodiment, in case a convex polygon as shown in FIG. 16A is to be determined with consideration given to the precision, characteristic, etc. of the stereo vision system 1, it will circumscribe the graphic shown in FIG. 16A as shown in FIG. 16B. Taking the precision, characteristic, etc. of the camera in consideration, a convex polygon inscribing the graphic shown in FIG. 16A, for



example, may of course be determined. Either of these methods may selectively be used depending upon the inclination of the plane and environmental conditions.

Also, in case a polygon including an input plane is determined by the convex hull algorithm as in step S2 in FIG. 15, a problem will arise with a non-polygonal stair. FIG. 19 explains the problem. FIG. 19A shows an input plane with "step 0" indicating a stair which is not any convex-polygonal. FIG. 19B shows the result of representation of "step0" in a polygonal form by a convex hull. As will be seen from FIG. 19B, there is a large difference of a non-convex portion from the desirable result of representation. In such a non-convex polygon, a polygon may be determined by smoothing with the close-gaps and fit-line-segments operations. Here will be explained the method of determining a polygon including an input plane by the smoothing with the close-gaps and fit-line-segments operations. FIG. 20 is a schematic illustration of the smoothing. FIG. 20A shows an input polygon which is a region including all supporting points determined to belong to one input plane (a group of distance data points in successive regions in the same plane), FIG. 20B shows a gaps-closed polygon resulted from smoothing the input polygon by removing discontinuous gaps from the latter, and FIG. 20C shows a smoothed polygon resulted from further smoothing of the polygon obtained as shown in FIG. 20B with the fit-line-segments operation. FIG. 21 shows an example of the program for determining a polygon including an input plane by smoothing with the close-gaps and fit-line-segments operations. In this illustration, there are

explained the close-gaps operation for removing discontinuous gaps from the polygon and fit-line-segments operation for further smoothing the resultant polygon with the fit-line-segments operation.

First, the close-gaps operation will be explained. Three successive ones of apexes defining a polygon are selected. If the central one of the apexes is largely distant from a straight line connecting the other two apexes, it is removed. This operation is repeatedly done with the remaining apexes of the polygon until there remain no more apexes to be processed in this way.

Next, the fitting-line-segments operation will be explained. Three successive ones of apexes defining a polygon are selected, and an approximate straight line fitted to the three apexes, respectively, and distances between the approximate straight line and apexes are determined by the least-squares method. All the distances from the approximate straight line are arranged in ascending order. If the distances are smaller than a threshold, the central one of the apexes is removed and the positions of the other two apexes are re-calculated based on the approximate straight line. This operation is repeatedly done with the remaining apexes until there remain no more apexes to be processed in this way.

Next, the operation made in step S3 of FIG. 15 will be described in detail. First, the stair climber 4 calculates the aforementioned stair parameters from the acquired polygon (in FIG. 16B, FIG. 20C). FIG. 22 is a schematic illustration for explaining how to calculate the stair parameters. As shown in FIG. 22A, the

acquired polygon 140 is taken as an area defined by points 141 to 147. It should be noted that the line forming the boundary at the front side, as viewed from the robot apparatus 201, of the polygon 140 is taken as the front edge FE while the line forming the inner boundary is taken as the back edge BE.

The width  $W$  of the stair tread can be  $W = d1 + d2$  ( $d1$  is the length of a line connecting the center  $C_{FE}$  of the front edge FE and reference point G and  $d2$  is the length of a line connecting the center  $C_{BE}$  of the back edge BE and reference point G).

Note that the reference point G may be generally the center of a plane as the tread. For example, the reference point G may be the central one of all the supporting points, center of gravity of the polygon 140 or the center of gravity of a safety zone 152 defined by connecting the end points of the front edge FE and back edge BE as shown in FIG. 22B.

With determining the length  $L$  and width  $W$  of the stair based on the front edge FE, back edge BE and reference point G determined as above, the stair climber 4 makes the operation in step S4. The stair length  $L$  of the stair may be, for example, a shorter one of the length of the front edge FE and that of the back edge BE or a longer one of the length of the front edge FE including the right and left margins which will be explained below and that of the back edge BE including the right and left margins.

Next, there will be explained how to calculate the margins in step S5. FIG.

23 is a schematic illustration for explaining a tread finally recognized tread and stair parameters. In this embodiment, it is assumed that margins  $M_1$  and  $M_2$  are provided at the left and right ends, respectively, of the safety zone 152 as shown in FIGS. 22B and 23 and the region 151 including the left and right margins  $M_1$  and  $M_2$  is finally recognized as a tread. To determine the margins  $M_1$  and  $M_2$ , the stair climber 4 first selects points defining a polygon, if any, protruding to outside the safety zone 152 defined by the front and back edges FE and BE. For determination of the right margin  $M_2$ , for example, the stair climber 4 will select points 142, 143 and 146 shown in FIG. 22A. For determining the right margin  $M_2$  adjacent to the right side of the safety zone 152, the stair climber 4 selects the point 142 farthest from the safety zone 152 and draws a perpendicular line from the point 142 down to the front and back edges FE and BE. The stair climber 3 will recognize the region 151 defined by the perpendicular line and front and back edges FE and BE as a tread. It should be noted that this margin may be determined just by drawing a line through the point 142 and across the front edge FE or back edge BE.

Assume here that the length of the left margin  $M_1$  in line with the front edge FE is  $lmf$  and that of the left margin  $M_1$  in line with the back edge BE is  $lbm$  as shown in FIG. 22B. Similarly, the length of the right margin  $M_2$  in line with the front edge FE and that of the right margin  $M_2$  in line with the back edge BE are  $rfm$  and  $rbm$ , respectively.

The effect of the recognition of stairs through determining the front and back edges FE and BE from a polygon as above will be explained herebelow. FIGS. 24A and 24B schematically illustrate two types of stairs, respectively. FIG. 24A shows stairs whose tread is rectangular as shown in FIG. 9 and 10, and FIG. 24B shows a spiral stair. In the spiral stair as shown in FIG. 24B, the back edge BE is not parallel with the front edge FE. So, an algorithm by which a rectangular area will simply be extracted from extracted planes may not be applied to such a spiral stair. Therefore, as in this embodiment, the robot apparatus can be caused to climb even such a spiral stair by determining a polygon from extracted planes and determining the front and back edges FE and BE on the basis of the polygon.

Next, the stair merging unit 6 shown in FIG. 14 will be explained in detail. The stair merging unit 6 is supplied with stair data (stair parameters) D3 extracted by the stair extraction unit 5 and merges the stair data D3 in a time sequence to estimate stair information more accurately and widely. For example, a robot apparatus having a narrow field of view will not be able to recognize all stairs at a time as the case may be. In such a case, old-stairs data on an old frame and new-stairs data on a new frame, for example, are searched for a set of stairs spatially overlapping each other and data on the stairs overlapping each other are merged together to define new virtual stairs. By repeating these operations until no more overlapping stairs exist, it is possible to recognize stairs accurately.

FIG. 25 shows a flow of operations made in stair-data merging in the stair

merging unit 6. First, the stair merging unit 6 is supplied with new-stairs data and old-stairs data in step S11, and merges all of them into one union in step S12. The stair merging unit 6 searches the union of merged stair data for stair data spatially overlapping each other in step S13. In case any set of overlapping stair data is found (Yes in step S14), the stair merging unit 6 will merge the stair data together and register the merged stair data into the stair data union in step S15. The stair merging unit 6 repeats the operations in steps S13 and S14 until no further set of spatially overlapping stair data exist (No in step S14), and outputs a finally updated stair data union as the stair data D4.

FIG. 26 is a schematic illustration for explaining the merging of stair data overlapping each other (as in step S13). FIG. 26 shows stair data ST11 and ST12 spatially overlapping each other. The stair merging unit 6 may judge whether the stair data spatially overlap each other on the basis of the difference in height (distance) between the two stair data at the reference point G in the two stair data ST11 and ST12 and the area where the treads including the right and left margins overlap each other. More specifically, in case the difference in height between the centers of gravity  $G_{11}$  and  $G_{12}$  of two stairs is smaller than a threshold ( $maxdz$ ) and the overlapping area is larger than a threshold ( $minarea$ ), the stair merging unit 6 can determine the treads indicated by the two stair data to overlap each other. In this case, the stair merging unit 6 merges the stair data ST11 and ST12 into stair data ST13 on a center of gravity  $G_{13}$  as shown in the lower illustration of FIG. 26.



For the merging, the stair merging unit 6 has the area of the outer frame including the stair data ST11 and ST12 as a step ST13, the area including the safety zone except for the right and left margins of the pre-merged stair data ST11 and ST12 as a new safety zone 165 after merged, and regions resulted from exclusion of the safety zone 165 from the stair data ST13 as the margins  $M_1$  and  $M_2$ . Thus, the stair merging unit 6 can determine the front and back edges FE and BE merged by the merged safety zone 165.

More specifically, the stair merging unit 6 determines opposite end points of the front edge FE in the merged stair data ST13 by comparing the right end point of the front edge FE of the stair data ST11 with that of the front edge FE of the stair data ST12 and also the left end point of the front edge FE of the stair data ST11 with that of the front edge FE of the stair data ST12, respectively. The stair merging unit 6 has the rightmost end point as a right end point 163 of the stair data ST13 and the leftmost end point as a left end point 164 of the stair data ST13. Also, the stair merging unit 6 determines the line position of the front edge FE of the stair data ST13 by comparing the front edge FE of the stair data ST11 with that of the stair data ST12. The stair merging unit 6 takes the line position nearest to the robot apparatus (near-side one) as the line position of the front edge FE of the stair data ST13. Similarly, the stair merging unit 6 selects the innermost line position of the back edge BE of the stair data ST13 as the back-edge line position, and the rightmost and leftmost end points as right and left end points 161 and 162,



respectively, of the back edge BE of the stair data ST13.

Note that the merging technique is not limited to the above. In this embodiment, merging is done with the field of view of the robot apparatus taken in consideration so that both the rectangular region defined by the front and back edges FE and BE and the merged data ST13 will be largest. However, in case the robot apparatus has a sufficiently large field of view or in case the distance data is sufficiently precise, the stair merging unit 6 may take a region resulted just from addition of two stair data as merged stair data. Also, the stair merging unit 6 can determine the reference point G after the merging by making weighted averaging correspondingly to the ratio between the numbers of supporting points included in the stair data ST11 and ST12, respectively.

Next, the stair climber 4 shown in FIG. 7 will be explained. The stair climber 4 is based on the stair data D4 merged and extracted by the stair extraction unit 5 and stair merging unit 6 to control the robot apparatus to climb stairs actually. The stair-climb control also includes the stair search.

The stair climbing implemented in this embodiment may include the following five operations (state machine):

#### 4-1 Search mode

To acquire range image by means of the stereo vision system installed in the head unit, the stair climber 4 has the robot apparatus turn the head unit thereof to take a look around in order to gather information on the surroundings.

## 4-2 Align mode

The robot apparatus stands in front of stairs, and moves over a predetermined fixed distance. FIG. 27 explains the align mode of operation. In FIG. 27, a region 170 is taken as a first one of the stair treads the robot apparatus recognizes. When in the align mode, the robot apparatus moves to a target position (will be referred to as "align position" hereunder) *align\_distance* from the center of the front edge FE of the tread 170 in a direction perpendicular to the front edge FE. In this case, it is assumed that the present position of the robot apparatus is a point 171 and the align position to which the robot apparatus is to move is a point 172. When the distance between these points 171 and 172 is larger than a predetermined threshold *max\_d* and the angle between the direction perpendicular to the front edge FE and a direction of the robot apparatus is larger than a predetermined threshold *max\_a*, the robot apparatus will start moving toward the align position 172. When the distance and direction are both reached, it is to be determined that the align mode has been completed. The above thresholds may be *max\_d* = 3 (cm) and *max\_a* = 2 deg., for example.

## 4-3 Approach mode

When in the approach mode of operation, the robot apparatus will move to just before the stairs. FIG. 28 is a schematic illustration for explaining the approach mode. As shown in FIG. 28, having completed the align mode by positioning itself in front of the stairs and moving to the align position 172

*align\_distance* from the starting position, the robot apparatus 201 will position itself in front of the center  $C_{FE}$  of the front edge FE of the tread 170 and move to an object position (will be referred to as "approach position" hereunder) a predetermined distance *approach\_x* distant from there to climb the stairs.

#### 4-4 Climb mode

The robot will climb the stairs on the basis of the stair data acquired through the aforementioned stair recognition. When the robot has arrived at a tread and viewed a next tread, it will continuously ascend or descend. The robot apparatus implements the stair climb mode by continuously moving on the stairs until no further stair exists.

#### 4-5 Finish mode

When the robot apparatus has ascended the stairs in the stair climb mode, it will make sure that it stands on the top stair and move to the center of the top stair. When the robot apparatus has descended the stairs in the stair climb mode or when it has descended the stairs ST1 as shown in FIG. 9 by turning around, for example, it will recognize that it has descended to the floor and exit the stair climb mode.

The stair climb mode will be described in further detail herebelow. FIG. 29 shows a flow of operations made in the stair climb mode. As shown in FIG. 29, the stair climber 4 once put into action will have the robot apparatus search stairs (in the search mode), move to a predetermined position opposite to the searched stairs (in the align mode) and approach the first one of the stairs (in the approach

mode) (in step S21). When the robot apparatus has succeeded in the sequence of search, align and approach (Yes in step S22), it will climb the stairs as will be described in detail later (in step S23) and informs the success in stair climbing. If the robot apparatus has failed in the above search, align and approach modes of operation (No in step S22), the stair climber 4 will exit the series of stair climbing operations with informing the failure in stair climbing. In this case, the stair climber 4 will have the robot apparatus repeat the operations starting at step S21.

Note that the sequence of search, align and approach in step S21 is the same as a control method used for the robot apparatus to normally arrive at an object or destination. More specifically, the following method is available, by way of example. FIG. 30 shows a flow of operations made in the search-align-approach sequence.

As shown in FIG. 30, when the search-align-approach sequence is started, the robot apparatus gets in a search mode (1) in step S32. When in the search mode (1), the stair climber 4 will have the robot apparatus turn the head unit thereof to take a look around in order to gather information on the environment as wide as possible. As a result, the stair climber 4 judges in step S32 whether the surroundings include any stairs the robot apparatus can climb. It should be noted here that when the height  $n_z$  of a plane included in the first one of the extracted stairs meets the condition that  $step\_min\_z < n_z < step\_max\_z$ , the stair climber 4 determines that the stairs are satisfactory for the robot apparatus to climb. When

there are stairs the robot apparatus can climb (Yes in step S32), the stair climber 4 will have the robot apparatus align itself to move over a predetermined distance (*align\_distance*) to the stairs in order to re-organize the stairs from a near position (in step S33), and thus re-recognize the stairs it is going to climb (step S34). The operation in step S34 is the "search mode (2)".

The stair climber 4 re-recognizes whether the stairs are satisfactory for the robot apparatus to climb (in step S35). When it has also succeeded in the search mode (2), it will make sure whether the robot apparatus is opposite to the re-recognized stairs and has moved over the align-distance to the align position, namely, whether it has succeeded in the align mode in step S33 (in step S36).

When there are stairs the robot apparatus can climb and the stair climber 4 has had the robot apparatus align itself to the stairs (Yes in steps S35 and S36), the stair climber 4 will have the robot apparatus get into the approach mode of operation in which it will move to the front edge of the first stair in step S37. On the other hand, if it has been determined in step S35 that the stairs are not satisfactory for the robot apparatus to climb, the stair climber 4 returns to step S31. If the robot apparatus has not succeeded in the align mode in step S36, the stair climber 4 will repeat the aforementioned operations in step S33 and subsequent steps.

Next, the stair climb mode of operation in step S22 will be described in detail. The stair climb mode consists of a process 1 in which the robot apparatus itself can recognize, from a tread on which it is currently standing, a one-step

higher or lower tread (will be referred to as "next tread" hereunder), a process 2 in which the robot apparatus cannot recognize the next tread from the currently standing tread but a two-steps higher or lower tread, and a process 3 in which the robot apparatus can recognize a plurality of stairs. FIGS. 31, 33 and 34 show flows of operations made in the above stair climbing processes 1 to 3 included in the stair climb mode of operation. It should be noted that in the following description, it is assumed that the tread (including the floor) on which the robot is currently moving is *step-0*, a next stair is *step-1*, a stair after next is *step-2* and a stair *m* steps higher or lower is *step-m*.

First, there will be explained the stair climbing process in which the tread *step-1* can be viewed and recognized. As shown in FIG. 31, when the stair climbing process 1 is started, the stair climber 4 will have the robot apparatus climb the stairs (climb operation (1)) (in step S41). In this climb operation (1), the stair climber 4 will select a mode of climbing by judging which the stair height  $n_z$  is, positive or negative, since the stair height  $n_z$  has been recognized in step S32. That is, when  $n_z < 0$ , the stair climber 4 will have the robot apparatus descend the stairs. When  $n_z > 0$ , the stair climber 4 will have the robot apparatus ascend the stairs. The value of a control parameter used for the climb operation varies depending upon whether the robot apparatus ascends or descends the stairs as will be described in detail later. That is, just by selecting a control parameter, the stair climber 4 can have the robot apparatus ascend or descend the stairs.



Then, the stair climber 4 judges, in step S42, whether the climb operation (1) has successfully been done or not. In case the climb operation (1) has successfully been done (Yes in step S42), the stair climber 4 will have the robot apparatus make a search operation (3). With this search operation (3), the robot apparatus moves the head unit having the stereo vision system installed therein to acquire distance data on the surroundings and extract a next stair. This search operation (3) is nearly similar to the search operation (2) in step S34 in FIG. 30.

When the search operation (3) has successfully been done as in step S35 in FIG. 30 (Yes in step S44), the stair climber 4 repeats the operation in step S41 again. If no stairs have been found (No in step S44), the stair climber 4 determines that the robot apparatus has ascended to the top tread or descended to the floor, and exits the search operation (3) with making the finish mode of operation (in step S45) such as movement to a predetermined position such as the tread center.

Next, the climb operation (1) in step S41 will be described in detail. FIG. 32 is a schematic illustration of stair treads the robot apparatus has recognized or is going to recognize. As shown in FIG. 32, it is assumed that the foot bases 121L and 121R of the currently moving robot apparatus are on the tread 181, for example. In FIG. 32, the robot apparatus will recognize, as a tread, the safety region defined by the front and back edges FE and BE and right and left margins  $M_1$  and  $M_2$  adjacent to the safety region. At this time, the robot apparatus has been able to



recognize the tread 182 of a next stair (*step-2*). It is assumed here that between the treads 181 and 182, there exists a gap 184 defined by the riser or the like.

With the climb operation (1), the stair climber 4 judges whether the robot apparatus can move from the tread 181 of the present tread (*step-0*) to the tread 182 of the next stair (*step-1*). When the tread 181 meets the following conditions, for example, the stair climber 4 determines that the robot apparatus can move so:

- 5-1 The tread 181 is sufficiently near a front edge  $FE_1$  of the tread 182 of the next stair (*step-1*) and the angular difference between these treads is smaller than a predetermined threshold
- 5-2 The tread 182 of the next stair (*step-1*) has sufficiently large dimensions
- 5-3 A distance *front\_x* from the front edge  $FE_1$  to the back edge of the foot bases 121L or 121R is larger than a control parameter *front\_x\_limit* in a specified mode of climbing
- 5-4 A distance *back\_x* from a back edge  $BE_1$  to the front end of the foot base 121L or 121R is larger than a control parameter *back\_x\_limit* in the mode of climbing

When the robot apparatus has been able to view the tread 183 of the next stair (*step-2*), the stair climber 4 can judge, based on a difference ( $z2 - z1$ ) between the height  $z1$  of the tread 182 of the next stair (*step-1*) at the reference point and height  $z2$  of the tread 183 of the stair after next (*step-2*) at the reference point,

which the climbing from the tread 182 of the next stair (*step-1*) to the tread 183 of the stair after next (*step-2*) is, ascending or descending. It should be noted that if the tread 183 of the stair after next (*step-2*) cannot be recognized, it may be considered that the current climbing is maintained.

In case the stair climber 4 has determined based on the above conditions 5-1 to 5-4 that the robot apparatus can climb the stairs, the robot apparatus can climb so. On the tread 181 of the present tread (*step-0*), the robot apparatus is aligned with the back edge  $BE_0$  of the tread 181. If the gap 184 between the treads 181 and 182 is large, the robot apparatus will align itself with the front edge  $FE_1$  of the tread 182 of the next stair (*step-1*), then move to the next stair and align itself with the back edge  $BE_1$  of the next stair. Then when in a next climb mode, the robot apparatus will align itself with the front end  $FE_2$  of the tread 183 of the stair after next (*step-2*), move to the tread 183 and align itself with the back edge  $BE_2$ . That is, the climb mode of operation is such that the robot apparatus aligns itself with the front edge FE of the tread of a next stair, climbs the stairs and aligns itself with the back edge BE of the tread of the stair to which it has moved, for example.

Also, when the gap 184 is small, the robot apparatus may align itself with only one of the back edge  $BE_0$  of the tread 181 of the present tread (*step-0*) and front edge  $FE_1$  of the tread 182 of a next stair under the assumption that these back and front edges are generally coincident with each other. Namely, the robot apparatus may also align itself with the back edge  $BE_0$  of the tread 181 of the

present tread, move to the tread 182 of a next stair and align itself with the back edge  $BE_1$  of the tread 182 or align itself with the front edge  $FE_1$  of the tread 182 of a next stair, move to the tread of the next stair and further align itself with the front edge  $FE_1$  of the tread 183 of the stair after next. In this case, the climb mode of operation is such that the robot apparatus moves to a next stair without aligning itself with the front edge FE of the next stair and then aligns itself with the back edge BE of the tread of the stair to which it has moved.

The climb operation (1) is applicable when the robot apparatus can view the tread of a next stair (*step-1*) to which it can move during stair climbing. For example, a bipedal walking robot apparatus should have installed therein the stereo vision system 1 capable of viewing the foot bases of the robot apparatus.

In the aforementioned stair climbing processes 2 and 3, the robot apparatus cannot view and recognize the tread of the next stair (*step-1*) from on the tread of the present tread (*step-0*) because of the limited angulation of the connection between the head and body units of the robot apparatus but can recognize the tread of the tread after next (*step-2*) or a m-steps higher or lower tread (*step-m*), which are different from the stair climbing process 1 in which the robot apparatus can view and recognize the tread of the next stair. The stair climbing processes 2 and 3 will be explained herebelow. First, in case the robot apparatus can recognize the tread of a stair after next (*step-2*), when the stair climbing process 2 is started, the robot apparatus will make the search operation (4) as above (step S51) as shown in

FIG. 32. With this search operation (4), the tread of the stair after next (*step-2*) is viewed and recognized by the stereo vision system 1. The search operation (4) is similar to the operation in step S43 except for recognition of the tread of the stair after next (*step-2*).

Then, the robot apparatus makes a climb operation (2) in step S52. The climb operation (2) is similar to the climb operation (1). Also in this operation, it can similarly be judged based on the height  $n_z$  of the tread of the next stair (*step-1*) which the climbing should be, ascending or descending. The tread of the next stair (*step-1*) is a tread viewable from the tread of a stair to which the robot apparatus has moved before it had moved to the tread of the present tread (*step-0*).

In case the climbing operation has successfully been done (Yes in step S53), if the result of search made in step S51 reveals that the two-steps higher or lower tread (*step-2*) has been extracted (Yes in step S54), the stair (*step-2*) is updated as a next stair (*step-1 = step-2*) in step S56. The stair climber 4 will repeat the operations in step S51 and subsequent steps. If the two-steps higher or lower tread (*step-2*) has not been extracted in the search operation (4) in step S51 (No in step S55), the robot apparatus will make the finish operation in step S55, and exit this series of operations.

Note that although the stair climbing processes have been explained under the assumption that a two-steps higher or lower tread can be viewed, the stair climbing processes can be executed in the similar manner even if a stair three steps

higher or lower can be viewed.

A stair climbing operation in which a tread a plurality of steps (m steps) higher or lower can be viewed and recognized will be explained herebelow as the stair climbing process 3.

In case the robot apparatus climbs stairs of which the first to n-th ones have been viewed and recognized, a search operation (5) is first done as shown in FIG.

34. The search operation (5) is basically similar to the search operation (4) in step S51 except that the treads of the stairs up or down to a stair m steps higher or lower (*step-m*) which can be recognized are to be viewed.

The climb operation (3) is effected for k steps in step S62. Also in this climb operation (3), a mode of climbing, ascending or descending, can be determined based on the difference in height between a plurality of treads having so far been viewed. Namely, when the height difference  $z_i - z_{i-1}$  between the i-th and (i-1)th treads is negative, the descending mode should be selected. When the height difference is 0 or positive, the ascending mode should be selected. Information on the tread on which the robot apparatus moves in this climb operation (3) is data having been viewed m steps before the robot apparatus is on the present tread.

When the robot apparatus has successfully climbed the stairs (Yes in step S63), the stair climber 4 judges in step S64 whether the robot apparatus has been able to view treads higher or lower than the present tread, namely, whether  $m + n >$

k. When  $m + n > k$  (Yes in step S64), the stair climber 4 will update *step*-( $k + 1$ ) to *step* -( $n + m$ ) as *step*-1 to *step*-( $m + n - k$ ) (where  $n = m$ ) in step S66, and repeat the operations in step S61 and subsequent steps. On the other hand, if  $m < 0$  (No in step S64), there exists no tread to which the robot apparatus moves. Therefore, the stair climber 4 exits the series of operations with executing the finish operation in step S65.

As having been described in the foregoing, the stair climber 4 can have the robot apparatus ascend or descend the stairs in the same manner just by selecting control parameters for use in the stair climbing processes 1 to 3, respectively, in the climbing operation. The control parameters used in the stair climbing are intended for limiting the position of the foot base of the robot apparatus in relation to the present tread.

FIG. 35A explains the relation between a tread the robot apparatus has recognized and the foot base, and FIG. 35B shows an example of the control parameters used in the climb operation.

The control parameters shown in FIG. 35A are as follows:

- step\_min\_z*     Minimum height difference (riser) between present and next treads, over which robot apparatus can climb
- step\_max\_z*     Maximum height difference (riser) between present and next treads, over which robot apparatus can climb
- ad(align\_distance)*     Distance between front edge FE and robot apparatus



which is in align position

*ax(approach\_x)* Distance between front edge FE and robot apparatus

which is in approach position

*front\_x\_limit* Limit value (minimal x-value) of distance between tread

front-edge FE and back end of foot base 121

*back\_x\_limit* Limit value (maximal x-value) of distance between tread

back-edge BE and front end of foot base 121

*back\_x\_desired* Desired value of distance between back edge BE and

front end of foot base 121

In ascending the stairs, if the riser is smaller than *step\_min\_z*, it will not be regarded that there exists any step. If the riser is larger than *step\_max\_z*, it will be determined that the robot apparatus cannot ascend the stairs. Similarly, in descending the stairs, if the riser is larger than *step\_max\_z*, it will not be regarded that there exists any step. If the riser is larger than *step\_min\_z*, it will be determined that the robot apparatus cannot descend the stairs.

The control parameter *align\_distance* is used in the align operation alone. It is used for starting the stair climbing process to ascend/descend the stairs, that is, for climbing the first stair. Similarly, the control parameter *approach\_x* is used in the approach operation alone. It is used for starting the stair climbing process to ascend/descend the stairs.

The control parameters *front\_x\_limit* and *back\_x\_limit* define the relation



between a tread the robot apparatus has recognized and the foot bases of the robot apparatus. When the distance between the tread back- and front-edges BE and FE and the end of the foot base, that is, a portion of the tread, which will not be stepped on under the foot steps of the robot apparatus that will move to the tread, is smaller than the values of these parameters, it will be determined that the robot apparatus will not be able to move to the tread or to make any next stair climbing even if it could have moved to the tread. It should be noted that when both the values of the parameters *front\_x\_limit* and *back\_x\_limit* are negative, the tread may be smaller than the foot bases. That is, it will be determined that the robot apparatus can ascend the stairs even if the tread is smaller than the foot bases.

The parameter *back\_x\_desired* is a distance between the back edge BE of the present tread and the front end of the foot base when the robot apparatus is to align itself with the back edge BE. As will be seen in FIG. 35B, when the robot apparatus is going to ascend the stairs, the parameter *back\_x\_desired* is normally a distance to a point short of the back edge BE, that is, a point 15 mm short of the back edge BE in this embodiment. When the robot apparatus is going to descend the stairs, the *back\_x\_desired* will be a distance to a point to which the foot base protrudes over the back edge BE, namely, a point 5 mm over the back edge BE in this embodiment. That is to say, ascending the stairs will need some distance or clearance for lifting and moving the foot to a next tread while descending the stairs will need no such distance, and a position protruding from the tread will enable the

robot apparatus to easily view and recognize a next tread and subsequent treads.

FIGS. 36 and 37 are traced drawings of the robot apparatus actually climbing the stairs on the basis of the control parameters shown in FIG. 35. FIG. 36 shows the robot apparatus ascending the stairs. The frames show the sequential motions of the robot apparatus ascending the stairs from the floor to the top stair. The frame No. 1 shows the robot apparatus in the search operation (1) in step S31. The frame No. 2 shows the robot apparatus in the align operation in step S33. The frame No. 3 shows the robot apparatus in the search operation (2) in step S32. The frame No. 4 shows the robot apparatus in the approach operation in step S37. The frame No. 5 shows the robot apparatus in the search operation (4) in step S51. The frame No. 6 shows the robot apparatus in the climb operation (2) in step S52. The frame No. 7 shows the robot apparatus still in the climb operation (2) in step S52 and also aligning itself with the back edge BE of the present tread. The frame No. 8 shows the robot apparatus in the search operation (4) in step S51. ... The frame No. 17 shows the robot apparatus having not viewed the next tread by effecting the search operation (4). The frame No. 18 shows the robot apparatus making the finish operation.

FIG. 37 shows the robot apparatus descending the stairs. As in the stair ascending sequence in FIG. 36, the robot apparatus repeats the search operation (as in frame Nos. 1, 4, 7, 10, 13 and 16) and climb operation (including the align operation as in frame Nos. 5, 6, 8, 9, 11, 12, 14 and 15), makes the finish operation

(as in frame No. 18) when no more treads are viewed, and exits the stair climbing.

Next, there will be illustrated and explained a variant of the embodiment of the present invention, having been described in the foregoing. The operations of ascending and descending the stairs have been explained in the foregoing.

However, the algorithm included in the control method according to the aforementioned embodiment can be applied to move the robot apparatus on a step formed from one stair or a one-step concavity, if any, when the step is smaller than a predetermined threshold as well as on the stairs as above.

First, there will be explained a motion of the robot apparatus to ascend a single step. FIG. 38 shows a geometric relation between the single step and the foot bases of the robot apparatus. A reference numeral 191 in FIG. 38 indicates a step whose height  $z_1$  from the floor surface ( $z = 0$ ) is 30. Here will be described a movement of the robot apparatus from the lower level of the step 191 which will be a next tread (*step 1*) to an upper level as in FIG. 38. It is assumed that the height  $Z_0$  of the moving plane at the lower level of the step 191 is  $z_0 = 0$  while the height  $z_2$  of the moving plane at the upper level of the step 191 is  $z_2 = 0$ . On the assumption that the robot apparatus is currently moving on the moving plane having the height  $z_0$ , since  $z_1 - z_0 = 30 > 0$ , the movement of the robot apparatus from on the step 191 to a next region can be determined to be "ascending" and also, since  $z_2 - z_0 = -30 < 0$ , such a movement can be determined to be "descending". Thus in the climb operation, the aforementioned control parameters should be

changed in value according to such determination. It should be noted here that in case the control parameters shown in FIG. 35B are adopted, values of both the parameters *front\_x\_limit* and *back\_x\_limit* for the robot apparatus to ascend the step are negative, which means that even if the foot bases 121 are protruding from the step 191 as shown in FIG. 38, the robot can be determined to be able to move.

Next, there will be explained a motion of the robot apparatus to step down into a single concavity. FIG. 39 shows a geometric relation between the single concavity and the foot bases of the robot apparatus. A reference numeral 192 in FIG. 39 indicates a concavity whose depth from the floor level ( $z = 0$ ) is  $z_1 = -30$ . Here will be described a movement of the robot apparatus from the lower level of the concavity 192 which will be a next tread (*step1*) to an upper level as in FIG. 39. It is assumed that the height  $Z_0$  of the moving plane at the lower level of the concavity 192 is  $z_0 = 0$  while the height  $z_2$  of the moving plane at the upper level of the concavity 192 is  $z_2 = 0$ . On the assumption that the robot apparatus is currently moving on the moving plane having the height  $z_0$ , since  $z_1 - z_0 = -30 < 0$ , the movement of the robot apparatus into the concavity 192 as a next region can be determined to be "descending" and also, since  $z_2 - z_1 = 30 > 0$ , the movement from the concavity 192 to a next region can be determined to be "ascending". Thus in the climb operation, the aforementioned control parameters should be changed in value according to such determination similarly to the step 191. It should be noted here that in case the control parameters shown in FIG. 35B are

adopted, values of both the parameters *front\_x\_limit* and *back\_x\_limit* for the robot apparatus to descend into the concavity are positive, which means that even if the foot bases 121 are smaller than the concavity 192 as shown in FIG. 39, the robot can be determined to be able to move.

In this variation, a plane which can be determined for the robot apparatus to move on, for example, a horizontal plane or the like, is extracted from the extracted planes and a stair tread is recognized based on a polygon including such plane. Then, the robot apparatus is made to climb the stairs on the basis of information on the front and back edges FE and BE of the stair tread and stair information including the height from the floor level. In the stair climbing, the tread is searched to see that the tread is satisfactory for the robot apparatus to move on, the robot apparatus is aligned with the front edge FE of the tread having been able to be searched or back edge BE of the present surface, and the control parameters are changed in value by judging, based on a height difference between the next and present treads, which one of the climbing modes the robot is going to be in, ascending or descending, whereby the robot apparatus can be made to climb normal stairs whose tread is rectangular as well as a spiral stair or the like and both the ascending and descending motions can be selected in the same manner just by changing the values of the control parameters. Therefore, the same controlling manner can be used to have the robot apparatus move on the stairs as well as a single step, single concavity or the like.

Also, since recognized stairs are merged in a time sequence in the stair recognition, the stair information can be recognized in a wide range even in a robot apparatus whose field of view is limited due to restrictions imposed on the location of a stereo vision system installed in the robot apparatus because object stairs are too large for the robot apparatus to climb. Also, even if a next tread cannot be viewed and recognized due to similar restrictions imposed on the location of the stereo vision system when the robot apparatus climbs the stairs on the basis of the stair information, the robot apparatus can climb the stairs on the basis of stair information already recognized by viewing.

The plane segmentation/extractor unit adopted in this variant can positively extract, by the scan line grouping method, a prevailing plane in the field of view as well as even a plurality of planes such as stairs. Also, in the line fitting effected for extraction of a plane or planes, the plane segmentation/extractor unit can permit a plane extraction robust against any measuring noises by fitting the lines adaptively to the distribution of distance data points.

FIG. 40 shows functional blocks of the plane segmentation/extractor unit used in this variant. As shown in FIG. 40, the plane detector, generally indicated with a reference numeral 100, includes a stereo vision system 1 as a distance data acquiring means for acquiring three-dimensional distance data, and a plane segmentation/extractor unit 2 to extract planes included in a range image formed from the three-dimensional distance data by the scan line grouping method. The



plane segmentation/extractor unit 2 includes a line fitting block 2a to select groups of distance data points estimated to be in one plane from the distance data points forming the range image and extract lines from each of distance data points in a group; and a region growing block 2b to grow one or a plurality of planar regions in the image from a line group including all the lines extracted by the line fitting block 2a. The region growing block 2b selects three arbitrary lines estimated to be in one plane from the line group and determines a reference plane on the basis of the three lines. Then, the region growing block 2b judges whether the lines adjacent to the selected three lines are in one plane. When it determines that the adjacent lines are in one plane, it will update the reference plane with the lines as region growing lines and grow the region of the reference plane.

The line fitting block 2a extracts a group of distance data points estimated to be in one plane in a three-dimensional space from each of the data trains along rows or lines of the range image and generates more than one line from the group of distance data points correspondingly to the distribution of the distance data points group. More specifically, if it is determined that the distribution is not uniform, it will be determined that the data point groups are not in one plane and the data point groups are segmented, it is judged whether or not each of the segmented groups of data points is distributed uniformly. These operations are repeatedly done. In case the distribution is determined to be uniform, lines are generated from the group of data points. The above operations are done as to all



the data trains to generate a group of generated lines D11, and the line group D11 is delivered to the region growing block 2b.

The region growing block 2b selects, from the line group D11, three lines estimated to be in one plane, and determines a plane which will be the seed of the reference plane on the basis of the three lines. By sequentially merging the planar region which will be the seed region with lines being in one plane in which the seed region also lies to grow the region, the range image is segmented into a plurality of planes to output a plane group D2.

Effecting the above operations when plane information is required for bypassing an obstacle or for stair climbing, the robot apparatus 201 acquires information on planes essential for the walking, such as stairs, floor surface, walls, etc.

To acquire such three-dimensional distance data by means of the stereo vision system 1, the stair ST2 should have a texture on the surface thereof. That is, since a parallax between two cameras is utilized to image a stair, if an object stair has no texture thereon, no parallax can be calculated and no accurate distance can be measured. Namely, the accuracy of distance data measurement by the stereo vision system depends upon a texture on an object stair. It should be noted that the "parallax" indicates a difference between images of a point in a space, formed on the left and right cameras (eyes), and varies depending upon the distance of the point from the cameras.

FIG. 41 explains a means for imparting such a texture, provided in the robot apparatus. As shown in FIG. 41, stereo cameras 11R and 11L forming the stereo vision system are provided in the head unit of the robot apparatus, and a light source 12 to emit infrared light, for example, is provided in the head unit. The light source 12 emits light to a stair ST3 having no texture on it or other object such as a thing having no texture or an insufficient texture or a wall to impart a random pattern PT. It should be noted that the means for imparting such a random pattern PT is not limited to any light source to emit infrared light if it can form such a random pattern PT to provide a range image but the robot apparatus itself may impart a texture on an object. However, the infrared light can impart a pattern an pattern which the human eyes cannot recognize but can be viewed by the CCD cameras or the like installed in the robot apparatus.

Next, the plane segmentation/extractor unit 2 included in the plane detector 100 will be explained. The plane segmentation/extractor unit 2 detects planes using the scan line grouping method in a variant of the aforementioned embodiment of the present invention shown in FIG. 42. In the scan line grouping method, an image 11 captured from a focus F is first processed along the row- or column-direction as shown in FIG. 42. Based on the fact that in a line-directional image train (image row), distance data points in one plane will form a straight line, there is generated a line including distance data points estimated to be in one plane. Namely, the scan line grouping method is to estimate and detect a plane in a group

of a plurality of thus acquired lines on the basis of the line groups determined to be in one plane.

FIG. 43 shows a flow of operations made in the plane detection by the scan line grouping method. As shown in FIG. 43, the plane segmentation/extractor unit 2 is first supplied with a range image in step S71. Then in step S72, the plane segmentation/extractor unit 2 determines lines from data points in each line- or column-directional image train in the range image, estimated to be in one plane. It extracts lines estimated to be in one plane from the line group to determine a plane including the lines in step S73. In step S73, a region which is to be the seed of the plane is selected. This selection is made under the condition that three lines including one of vertically adjacent rows (or horizontally adjacent columns) should be in one plane. The plane in which the seed region defined by the selected three lines is taken as a reference plane, and a plane is determined by averaging the three lines. Also, a region defined by three lines is taken as a reference planar region.

Then, the plane segmentation/extractor unit 2 compares, in clearance, the straight line including the line- or column-directional image trains adjacent to the selected seed region and the reference plane to judge whether they are in one plane. If they are determined to belong to the same plane, the plane segmentation/extractor unit 2 adds the adjacent line to the reference plane area (region growing), updates the reference plane as a plane including the added line (plane updating), and repeats these operations until no more lines in one plane exist

in the data train adjacent to the planar region. Then, it searches seed regions, and repeats the plane updating and region growing until no seed region (three lines) exist. Finally, the plane segmentation/extractor unit 2 couples ones of the thus acquired region groups, which form together the same plane. In this variant, there is provided step S74 in which there will be done a plane recalculation to determine a plane again by calculating the line groups in the acquired plane except for ones more than a predetermined threshold off the plane. The plane thus determined is taken as a final one. This will be described in further detail later.

Note here that the process to detect lines from the three-dimensional distance data and take, as one plane, a region collecting the lines per each same plane is the plane detection by the conventional scan line grouping method. However, the line fitting method in step S72 in this variant is different from the conventional one. That is, even when it is intended to generate lines fitting the distance data points as much as possible by determining the lines from the distance data points as above, problems such as over-segmentation and under-segmentation will arise unless the threshold is changed correspondingly to the accuracy of the distance data. On this account, a method of changing the threshold adaptively to the accuracy of distance data and noises is introduced into the line fitting in this variant.

The plane detection by the scan line grouping method will be described in further detail with reference to FIG. 43. The plane segmentation/extractor unit 2 includes a line fitting block 2a which is supplied with three-dimensional range

image from the stereo vision system 1 and detects lines estimated to be in one plane in a three-dimensional space. To avoid the problem such as the over-segmentation or under-segmentation, that is, recognition of a plurality of planes as a single plane or recognition of a single plane as a plurality of planes, an algorithm (adaptive line fitting) is introduced to fit lines adaptively to the distribution of data points. The plane segmentation/extractor unit 2 also includes a line fitting block 2a. The adaptive line fitting is such that the line fitting block 2a first extracts a line as a first one with a relatively large threshold and then analyzes the distribution of the data point groups forming a line, as a second one, determined by the least squares method which will be described in detail later on the basis of the extracted first line. That is to say, it is roughly estimated whether the data point groups are in one plane, to thereby extract the data point groups, and it is analyzed whether the data points are unevenly distributed in the extracted data point groups, to thereby re-estimate whether the data point groups are in one plane.

This variant uses an algorithm in which when the result of the data point distribution analysis is such that the data point groups are in a zig-zag shape which will be described in detail later, it is determined that the distribution is uneven, and the data point groups are segmented, and these operations are repeatedly done to extract lines adaptively to noises included in the data point group.

FIG. 44 shows a flow of operations made in the line fitting block 2a, namely, it shows the operations in step S72 in FIG. 43 in detail. First, the line fitting block

2a is supplied with distance data. It extracts a group of data points estimated to be in one plane in the three-dimensional space in the line-directional image trains (data point train), for example, of the supplied distance data. The group of data points estimated to be in one plane in the three-dimensional space may be a set of data points whose mutual distance in the three-dimensional space is less than a predetermined threshold, for example, 6 cm, and is extracted as a data point group  $(P[0, \dots, n-1])$  in step S81. Then, it is checked whether the number of samples  $n$  included in the data point group  $(P[0, \dots, n-1])$  is larger than a minimum necessary number of samples  $min\_n$  (in step S82). When the number of samples  $n$  is smaller than the minimum necessary number  $min\_n$  (Yes in step S82), the process is ended with outputting an empty set as the result of detection.

On the other hand, when the number of samples  $n$  is larger than the minimum necessary number  $min\_n$  (No in step S82), a line (chord)  $L1$  connecting one end  $P[0]$  of the data point group  $(P[0, \dots, n-1])$  and other end  $P[n-1]$  is generated as a first line. Then, a data point whose distance from the line  $L1$  is searched as a point of interest  $brk$  from the data point group  $(P[0, \dots, n-1])$ , and the distance  $dist$  is calculated (in step S83). If the maximum distance  $dist$  is larger than a threshold  $max\_d$  for segmentation of the data point group (Yes in FIG. 84), the data point group  $P[0, \dots, n-1]$  is segmented by the point of interest (break point)  $brk$  into two data point groups  $P[0, \dots, brk]$  and  $P[brk, \dots, n-1]$  (in step S88).



On the other hand, if the maximum distance *dist* is smaller than the threshold *max\_d* for segmentation of the data point group (No in step S84), the least squares method which will be described in detail later is used to determine an optimum-line equation *line* from the data point group ( $P[0, \dots, n-1]$ ) (in step S85) and a line L2 defined by the equation *line* is generated as a second line. Then, it is examined in step S86 whether the data point groups ( $P[0, \dots, n-1]$ ) are in a zig-zag shape in relation to the line L2. When the data point groups ( $P[0, \dots, n-1]$ ) are not in the zig-zag shape (No in step S86), the process is ended with adding the determined line equation *line* to a line-extraction results list (in step S87).

If it has been determined in step S86 that the line determined in step S85 is in the zig-zag shape (Yes in step S86), the line fitting block 2a goes to step S88 as in step S84, and will segment the data point group by the point of interest *brk* whose distance *dist* has been determined in step S83 into two data points  $P[0, \dots, brk]$  and  $P[brk, \dots, n-1]$ . If the two data point groups are acquired in step S88, the line fitting block 2a will recursively process the two data point groups by making the operations in step S81 and subsequent steps. The line fitting block 2a will repeat the operations until all the data point groups will not be segmented any more, that is, until all the data point groups have been processed through step S87, whereby a line-extraction results list having all the lines registered there is provided. With such operations, the influence of noises can be removed from the data point group  $P[0, \dots, n-1]$  to accurately extract a group of lines.



Note that although it has been described above that the line  $L1$  connecting the end points of the data point group  $P[0, \dots, n-1]$  is generated in step S83, the line  $L1$  may be determined by the least-squares method from the data point group  $P[0, \dots, n-1]$  correspondingly to the distribution and nature, for example, of the data point group  $P[0, \dots, n-1]$ . Also, although the point of interest  $brk$  is taken as a point whose distance to the line  $L1$  connecting the end points, it may be a point whose distance to a line determined by the least-squares method as above is largest. However, in case a point whose distance to a line determined by the least-squares method is largest is taken as the point of interest  $brk$  or in case there is a plurality of points whose such distance is larger than the threshold  $max\_d$  for segmentation of the data point group, the data point group  $P[0, \dots, n-1]$  may be segmented by all such points or by more than one selected from the points.

Next, the least-squares line fitting by the least-squares method as in step S85 will be described. There will be explained a method of determining an equation for a line most fit to the data point groups when  $n$  data point groups  $P[0, \dots, n-1]$  are given. A model of the line equation is given by the following equation (1):

$$x \cos \alpha + y \sin \alpha + d = 0 \quad \dots(1)$$

In this case, the sum of differences between a line equation model and data points at one point  $(x_i, y_i)$  of  $n$  data point groups  $P[0, \dots, n-1]$  can be given by the following equation (2):

$$E_{fit} = \sum_i (x_i \cos \alpha + y_i \sin \alpha + d)^2 \quad \dots (2)$$

A straight line most fit to the data point groups can be determined by minimizing the sum of the differences given by the above equation (2). The terms  $\alpha$  and  $d$  for minimizing the equation (2) can be determined as given by the following expression (3) using a mean value and variance-covariance matrix of the data point groups P:

$$\alpha = \frac{1}{2} \tan^{-1} \frac{-2S_{xy}}{S_{y^2} - S_{x^2}}, \quad d = -(\bar{x} \cos \alpha + \bar{y} \sin \alpha) \quad \dots (3)$$

where

$$\begin{aligned} \bar{x} &= \frac{1}{n} \sum_i x_i & \bar{y} &= \frac{1}{n} \sum_i y_i \\ S_{x^2} &= \sum_i (x_i - \bar{x})^2 & S_{y^2} &= \sum_i (y_i - \bar{y})^2 \\ S_{xy} &= \sum_i (x_i - \bar{x})(y_i - \bar{y}) \end{aligned}$$

Next, the zig-zag shape discrimination effected in step S86 will be explained. In this zig-zag shape discrimination method, when  $\underline{n}$  data point groups  $P[0, \dots, n-1]$ , straight line  $Line(\alpha, d)$  and  $x \cos \alpha + y \sin \alpha + d = 0$  are given, it is

discriminated whether the data point groups  $P[0, \dots, n-1]$  intersect the straight line *Line* as shown in FIG. 45A or whether the data points are evenly distributed under the influence of noises for example, as shown in FIG. 45B. Basically, the data point groups  $P[0, \dots, n-1]$  appearing successively on the straight line *Line* is counted. When a larger number of data point groups  $P[0, \dots, n-1]$  than a specific number appear successively, it can be determined the data point groups form a zig-zag shape. In case the data point groups intersect the straight line *Line* as in FIG. 45A, the data point group  $P[i]$  have to be segmented for determination of a straight line *Line* more fit to the data point groups  $P[0, \dots, n-1]$ . FIG. 46 shows a flow of operations made in the zig-zag shape discrimination method.

First, the line fitting block 2a is supplied with the data point groups  $P[0, \dots, n-1]$  and straight line *Line* ( $\alpha, d, \sigma$ ) in step S90. The " $\sigma$ " indicates a standard deviation of the point train. Next, the line fitting block 2a judges whether the standard deviation  $\sigma$  is larger than a threshold  $th\_ \sigma$ . If the standard deviation  $\sigma$  is smaller the threshold  $th\_ \sigma$  (No in step S91), the line fitting block 2a terminates the judgment in order avoid the influence caused by the detection of a floating point calculation error of the CPU. The discrimination is continuously done only when the standard deviation  $\sigma$  is larger than the threshold  $th\_ \sigma$ . Next, the line fitting block 2a judges, based on *sing* (*sdist*( $P[0]$ )), on which side of the straight line there is the first data point  $P[0]$  of the data point groups  $P[0, \dots, n-1]$ , places the result of judgment in *val<sub>0</sub>* and sets the count of a counter which counts successive data point

groups on the same side as  $val_0$  (which counter will be referred to as "successive point counter; the count of this counter will be referred to as "*count*" hereunder) to one (1) in step S92. It should be noted that "*sign(x)*" is a function to repeat the sign (+ or -) of the value of "*x*", and "*sdist(i)*" indicates a positive or negative distance from the *i*-th data point in the straight line *Line* determined by calculation of  $P[i].x\cos\alpha + P[i].y\cos\alpha + d$ . That is, a sign "+" or "-" is placed in  $val_0$  depending upon the side of the line *Line* on which the data point  $P[0]$  is.

Next, a counter to count data points (will be referred to as "data point counter" hereunder; count of this data point counter be referred to as "*count i*" hereunder) is set to "1" (count "i") in step S93. When the *count i* of the data point counter is smaller than the number of data *n* (Yes in step S94), the line fitting block 2a judges, based on  $\text{sing}(\text{sdist}(P[i]))$ , on which side of the straight line the data point  $P[i]$  on which a next data (will be referred to as "*i*-th data" hereunder) is, and places the result of judgment in *val* (in step S95). Then, the line fitting block 2a compares  $val_0$  determined in step S92 and *val* determined in step S95. When  $val_0$  and *val* are different from each other (No in step S96), the line fitting block 2a places *val* in  $val_0$ , places "1" in the count *count* of the successive point counter (in step S98), increments the count "i" of the data point counter (in step S100) and then repeats the operations in step S94 and subsequent steps.

On the other hand, when  $val_0$  and *val* are equal to each other (Yes in step S96), the line fitting block 2a determines that the data points  $P[i-1]$  and  $P[i]$  are on

the same side of the straight line *Line* and increments the count *count* of the data point counter by one (1) in step S97. Further, the line fitting block 2a judges whether the count *count* of the successive point counter is larger than the minimum data point count *min\_c* for determination of the data point groups to be in the zig-zag shape (in step S99). When the counter is larger than the minimum data point count *min\_c* (Yes in step S99), the line fitting block 2a determines the data point groups to be in the zig-zag shape, and exits the process with outputting a signal TRUE. On the other hand, when the count *count* of the successive point counter is smaller than the minimum data point count *min\_c* (No in step S99), the line fitting block 2a goes to step S100 where it will increment the *count i* of the data point counter (in step S100), and repeats the operations in step S94 and subsequent steps.

Then, the line fitting block 2a repeats the operations in step S94 and subsequent steps until the *count i* of the data point counter reaches the data number *n*, and it will exit the process with outputting a signal FALSE when a state of *count i*  $\geq n$ .

When *n* data point groups  $P[0, \dots, n-1]$  and straight line *Line* ( $\alpha, d$ ):  $x\cos\alpha + y\cos\alpha + d = 0$  are given, the line fitting block 2a can judge whether the group of data point group intersects the straight line *Line* in a zig-zag shape. Thus, the line fitting block 2a can judge in step S86 whether the data point groups should be segmented. When the line fitting block 2a has determined that the group of data

point group intersects the straight line determined by the least-squares method in the zig-zag shape, it determines that the data point groups should be segmented, and will go to step S88 where it will be able to segment the data point groups by the point of interest *brk* as a break point. It should be noted that the operations made in steps S91 and S100 may be expressed as in FIG. 47.

Also, the zig-zag shape discrimination can be effected by hardware as well as the CPU. FIG. 48 is a block diagram of a zig-zag shape discriminator. As shown in FIG. 48, the zig-zag shape discriminator, generally indicated with a reference numeral 20, includes a direction discriminator 21 which is supplied with  $n$  data point groups  $P[0, \dots, n-1]$ , judges on which side of the straight line *Line* each of the data point  $P[i]$  is and outputs the result of judgment *val*, a delay unit 22 for comparison of the data one after each data point  $P[i]$  with the result of judgment from the direction discriminator 21, a comparator 23 to compare the result of direction discrimination *val* at the data point  $P[i]$  and result of direction discrimination *val<sub>0</sub>* at the data point  $P[i-1]$ , a successive point counter 24 to increment the count when the comparator 23 provides a result that  $val = val_0$ , and a comparator 25 to compare the count *count* of the successive point counter 24 with the minimum data point count *min<sub>c</sub>* read from a minimum data point count storage unit 26.

The zig-zag shape discriminator 20 constructed as above functions as will be described below. The direction discriminator 21 determines a straight line *Line* by

the least-squares method from the data point groups  $P[0, \dots, n-1]$ , determines a positive or negative distance between each data point  $P[i]$  and straight line *Line* and outputs the sign, positive or negative of the distance. When supplied with the positive or negative sign of the distance from the data point  $P[i-1]$  to the straight line *Line*, the delay unit 22 stores the data until it is supplied with the positive or negative sign of the data point  $P[i]$  one after the data point  $P[i-1]$ .

The comparator 23 compares the sign, positive or negative, of the data point  $P[i]$  and that of the data point  $P[i-1]$ . When the signs are the same, the comparator 23 will output a signal for incrementing the count *count* of the successive point counter 24. When the signs are different from each other, the comparator 23 will output a signal for placing one (1) in the count *count* of the successive point counter 24. The comparator 25 compares the count *count* and minimum data point count *min\_c*. When the minimum data point count *min\_c* is larger than the count *count*, the comparator 25 will output a signal indicating that the data point groups  $P[0, \dots, n-1]$  are in a zig-zag shape.

Next, the region growing block 2b shown in FIG. 40 will be explained. The region growing block 2b is supplied with line groups determined by the line fitting block 2a, judges, by fitting the point train to a plane (plane fitting) in which plane each of the lines is, and separates a region formed from the supplied line groups into a plurality of planes (planar regions). For the separation of a region into a plurality of planes, the following technique is adopted.



First, the region growing block 2b searches the supplied line groups for three adjacent lines estimated to be in one plane. A plane determined based on these three lines (reference plane) is the seed of planes. A region including the three lines is called "seed region". The region growing block 2b judges, by the plane fitting, whether the lines adjacent to the seed region is included coplanar with the reference plane. When the lines adjacent to the seed region is included in one plane, the region growing block 2b will add the lines as region-growing lines to the seed region to grow the region, while recalculating the reference-plane equation for the region-growing lines as well. With the above operations, all the lines are distributed in a region (plane).

FIG. 49 is a schematic illustration for explaining the region growing. When stairs 31 defined by a plurality of planes exist in an image 30 as shown in FIG. 49, it is meant that three lines 32a to 32c indicated each with a thick line, for example, have been selected. A region defined by these lines 32a to 32c is a seed region. First, one plane (reference plane) P is determined based on these three lines 32a to 32c. Next, lines lying coplanar with the plane P are selected based on a data train 33 or 34 adjacent, outside the seed region, to the lines 32a to 32c lying at the outermost side of the seed region. It is assumed herein that a line 33a is selected. Then, a plane P' including the group of these four lines is determined to update the reference plane P. Further, when a line 34a is selected, a plane P'' including the group of these five lines is determined to update the plane P'. With these

operations being repeatedly done, the second tread of the stairs 31 is determined as a plane 45 shown as enclosed with a dash line. Thus, the region growing process is effected until no further lines exist for addition to a selected seed region. When there is no further line, the image 30 is searched for three lines which are to form a seed region and the region growing process is repeated. With these operations being done repeatedly, the operation in step S3 in FIG. 43 is repeated until there are no further three lines for a seed region.

Next, there will be explained the plane fitting technique for estimation of a plane equation formed from the group of data points  $P[0, \dots, n-1]$ , the seed region selecting technique for selection of a seed region on the basis of the estimated plane equation, region growing technique for making a seed region larger, and the post-processing technique for recalculating regions except for those having large errors on the basis of the estimated plane equation.

The point  $P$  in a three-dimensional space is given by  $P = (x_i, y_i, z_i)$ , and the plane equation is expressed with its normal vectors  $n(n_x, n_y, n_z)$  and a non-negative constant  $d$  as given by the following equation (4):

$$xn_x + yn_y + zn_z + d = 0 \quad \dots(4)$$

Note that a plane passing through a focus of a stereo camera cannot be viewed by the stereo camera. That is to say, since the plane does not pass by the focus, the term "d" in the above equation (4) may be  $d \neq 0$ . Therefore, the

least-squares method can be used to determine the plane as a value which minimizes the value as given by the following equation (5):

$$fit(\mathbf{n}, d) = \sum_i (\mathbf{p}_i^T \mathbf{n} + d)^2 \quad \dots (5)$$

The optimal solution is determined as  $\mathbf{n} = \mathbf{m}/\|\mathbf{m}\|$  and  $d = -1/\|\mathbf{m}\|$  where  $\|\cdot\|$  is a vector value and  $\underline{\mathbf{m}}$  is a solution of a linear system easily obtainable as given by the following equation (6-1) using the Cramer's rule which is to solve the simultaneous equation by a determinant:

$$\mathbf{A} \cdot \mathbf{m} = \mathbf{b} \quad \dots (6-1)$$

where

$$\mathbf{A} = \sum_i \mathbf{p}_i \mathbf{p}_i^T, \quad \mathbf{b} = \sum_i \mathbf{p}_i \quad \dots (6-2)$$

Even if new data points have been added or data points have been deleted, the above solution permits to recalculate the plane parameters just by updating the values  $\mathbf{A}$  and  $\underline{\mathbf{b}}$  in the equation (6-2). Further, in the line fitting in this variant, two moments (primary moment: average; geometrical moment: variance)  $E(\mathbf{p})$  and

$E(pp^T)$  of the group of  $\underline{n}$  data points are already known, and they can be used to update the values  $\underline{A}$  and  $\underline{b}$  as in the following expression (7) for use in plane updating of the group of  $\underline{n}$  data points:

$$\underline{A} \leftarrow \underline{A} + nE(pp^T) \quad , \quad \underline{b} \leftarrow \underline{b} + nE(p) \quad \dots(7)$$

Also, once the parameters  $\underline{n}$  and  $\underline{d}$  of a plane are calculated, the root mean square (RMS) residual of the plane equation indicating the extent of deviation of the group of  $\underline{n}$  data points from the plane equation (will be referred to as "*rms*" hereunder) can be calculated using the following equation (8). Also in this case, the two moments of the  $\underline{n}$  data points can be used to calculate the following equation (8):

$$rms(p_1 \dots p_n) = \sqrt{\frac{1}{n} \sum_i (p_i^T \underline{n} + d)^2} > th\_rms \quad \dots(8)$$

As will be known from the above equation (8), when each data point is in a determined plane, the *rms* ( $p_1, \dots, p_n$ ) of the plane equation will be zero (0). If the value is smaller, it is meant that each data point is well fit to the plane.

Next, there will be described the technique for searching a seed region and technique for making the seed region larger while updating a plane. FIG. 50

shows a flow of operations made in seed-region search and region growing processes. As shown in FIG. 50, to search a seed region, three lines  $l_1$ ,  $l_2$  and  $l_3$  to which the line- or column-directional data train having been used in the line fitting and whose pixel positions in the lines  $l_1$  and  $l_2$  and lines  $l_2$  and  $l_3$  overlap each other in a direction perpendicular to the data train are searched in step S101. Each of the data points has an index indicating a pixel position in an image. Lines in a line-directional data train, for example, are compared based on their own indexes to see whether they overlap each other in the direction of the data train. When such lines have successfully been found out (Yes in step S102), the above equation (7) is used to calculate the above equation (6-1). Thus, the plane parameters  $\underline{n}$  and  $\underline{d}$  can be determined, and they are used to calculate root mean square residuals ( $l_1, l_2, l_3$ ) of the plane equation as given by the above equation (8) (in step S103). In case the root mean square residuals  $rms(l_1, l_2, l_3)$  of the plane equation is smaller than a predetermined threshold  $th_{rms}l$  such as 1 cm, the three lines are selected as a seed region in step S104. When the root mean square residuals  $rms(l_1, l_2, l_3)$  of the plane equation is larger than the predetermined threshold  $th_{rms}l$ , the region growing block 2b returns to step S101 where it will search lines which meet the above requirements. Also, the lines selected as the seed region are excluded from the line groups list so that they will not be used in any other plane growing process or the like.

The seed region thus selected is made by the scan line grouping method to

grow. That is, there are first searched lines which are candidate ones to be added to the seed region in step S105. It should be noted that this region includes a seed region already updated as well. The candidate lines are a line ( $l_4$ ) adjacent to the line  $l_1$ , for example, included in the seed region. The pixel positions in these lines have to overlap each other as above. When such lines have successfully been searched (Yes in step S106), the root mean square residual  $rms$  ( $l_4$ ) of the plane equation is calculated, and it is judged whether the  $rms$  is smaller than a predetermined threshold  $th_{rms}$  in step S107. When the  $rms$  ( $l_4$ ) is smaller, the plane parameters are updated in step S108 and the operations in step S105 and subsequent steps are repeated. The operations are repeated until no further candidate lines exist. When there exist no further candidate lines (No in step S106), the region growing block 2b returns to step S101 and searches a seed region again. When no seed region is included in the line group (No in step S102), the region growing block 2b exits the process with outputting the plane parameters having so far been acquired.

In this variant, the above equation (8) is used for searching a seed region and judging whether three lines are in one plane and also used to judge, in growing the region, whether the lines are in the reference plane or a plane resulted from updating of the reference plane. More specifically, only when the root mean square residual  $rms$  of the plane equation is less than the predetermined threshold ( $th_{rms}$ ), the lines (group) are estimated to be in one plane and a plane is

recalculated as a plane including the lines. By judging, based on the root square mean residual *rms* of the plane equation, whether the lines are in one plane, a plane can accurately be extracted even when the plane includes small steps robust against noises. The reason will be explained below.

FIG. 51 schematically illustrates examples equal in distance between an end point and straight line to each other but different in root mean square residual *rms* of the plane equation from each other. As disclosed in Non-Patent Publication 4, in region growing with a straight line being coplanar with a plane P when the distance D between an end point of the straight line (line segment) of interest and the plane P is smaller than a predetermined threshold, a straight line La intersecting the plane P (as in FIG. 51A) and straight line Lb parallel with the plane P and a predetermined distance off the latter (as in FIG. 51B) will be used to update the plane P. The root mean square residual *rms* of the plane equation is determined. The root mean square residual *rms* (La) of the plane equation, determined based on the straight line La in FIG. 51A is smaller than the root mean square residual *rms* (Lb) of the plane equation, determined based on the straight line Lb in FIG. 51B. That is, as shown in FIG. 51A, when the straight line La and plane P intersect each other, the root mean square residual *rms* of the plane equation is relatively small and influenced by noises in many cases. On the other hand, as shown in FIG. 51B, the root mean square residual *rms* of the plane equation is large and it is highly probable that the straight line Lb is not coplanar with the plane P but with another



plane  $P'$ . Therefore, in case it is necessary to accurately determine a plane from an environment including a plurality of planes, it will be preferable to calculate a root mean square residual  $rms$  of a plane equation and determine lines to be in one plane when the  $rms$  value is smaller than a predetermined threshold as in this variant. It should be noted that when the distance between the end point of a line and a plane is smaller than a predetermined threshold, the line may be included in the plane as in the conventional technique. The former and latter techniques may be used in combination depending upon the environment and nature of distance data.

Also, once the plane parameters  $\underline{n}$  and  $\underline{d}$  are calculated, the root mean square residual  $rms$  of the plane equation can easily be calculated by the aforementioned equation (8) by updating the plane equation on the basis of the values of two moments determined during line fitting of a group of data points.

Also, the aforementioned selection of a seed region can be expressed as shown in FIG. 52. In FIG. 52, "overlap( $l_j, l_k$ )" is a function for outputting "true" when a position between end points in straight line vectors  $l_j$  and  $l_k$  included in each image row overlaps the straight line vectors in a position of intersecting the line vectors. Also, "fitplane( $l_1, l_2, l_3$ )" is a function for determining a solution of  $A\mathbf{m} = \mathbf{b}$  by the aforementioned equations (4) to (7), calculating the plane parameters  $\underline{n}$  and  $\underline{d}$  and fitting line vectors  $l_1, l_2$  and  $l_3$  to the plane by the terms  $\mathbf{A}$  and  $\mathbf{b}$  calculated by the equation (8).

" $rms(l_1, l_2, l_3)$ " is a function for calculating, using the equation (6), the value

of root mean square residual *rms* of the plane equation as to all the three straight lines. Also, "*remove*( $l_1, l_2, l_3$ )" means removal of straight lines  $l_1, l_2$  and  $l_3$  selected as forming a seed region from *lines*[*i*], *lines*[*i*+1]  $l_2$  and *lines*[*i*+2], respectively, which removal will prevent these straight lines from being used again for calculation.

Also, the region growing may be expressed as in FIG. 53. In FIG. 53, the terms *A* and *b* are a matrix and vector, respectively, included in the aforementioned equation (6-1). Also, "*add*(*A*, *b*, 1)" is a function for adding the moment of a straight line *Line* to *A* and *b* using the equation (8). "*Solve*(*A*, *b*)" is used to determine the value *m* which meets  $Am = b$  and calculate the plane parameters *n* and *d* on the basis of the equations (4) to (7). "*select*(*open*)" is a function for selecting one element arbitrarily from "*open*". Also, "*index*( $l_1$ )" is a function for returning the index of " $l_1$ " in the image column or row. Also, "*neighbor*(*index*)" is a function for returning an index adjacent to a given index, for example, {*index*-1, *index*+1}.

Also, in this variant, the region growing is done in step S73 in FIG. 43 to update the plane equation, and then the plane equation is recalculated (which is a post-processing) in step S74, as having previously been described. In this recalculation, a displacement, from the plane, of the distance data or lines estimated to be in a plane defined by the plane equation finally acquired through the aforementioned updating, for example, and the plane equation is updated again

with exclusion of the distance data or lines more than a predetermined value off the plane, whereby the influence of noises can further be reduced.

Next, the operation in step S74 will be described in further detail. Here will be explained a technique for recalculation of the plane equation in two steps. First, when data points nearer to a plane adjacent to a plane in which the data points currently exist than the latter plane are detected at distance data points (pixels) at the boundary of each plane detected in step S73, the data points are included into the adjacent plane. Also, in case there can be detected data points not in any of plane but through which there passes a plane whose distance to the data points is smaller than a relatively large threshold such as 1.5 cm, the data points in question are included into that plane. These operations can be effected by searching data points near the boundary of each planar region. After completion of these operations, the plane equation is recalculated.

Next, in case the distance between each data point and the plane exceeds a relatively small threshold such as 0.75 cm at a location near the boundary of each region of a plane recalculated as above, such data points are discarded. Thereby, the planar region will be somewhat smaller but a more accurate plane can be determined. The deletion of distance data points and recalculation of a plane are repeatedly done, whereby an extremely accurate plane can be determined.

Next, the result from each of the above operations will be described. FIG. 54A is a schematic illustration of a floor surface viewed from the robot apparatus

being in upright position, and FIG. 54B shows three-dimensional distance data with the vertical axis taken as  $x$ , horizontal axis as  $y$  and the density of each data point taken as  $z$ , and straight lines detected from a group of data points included in a line-directional pixel train and determined in the line fitting process on the basis of to lie in one plane. FIG. 54C shows a planar region determined by the region growing from the group of straight lines shown in FIG. 54B. As seen in these drawings, only one plane (floor surface) exists in the field of view of the robot apparatus, namely, all the floor surfaces are detected as the same plane.

FIG. 55 explains the difference in result of performance between the plane segmentation/extraction method according to the variant and conventional plane segmentation/extraction method when a step is placed on a flat surface. As shown in FIG. 55A, a step ST3 is placed on the floor F. FIG. 55B shows experimental conditions. In case the distance between a point of interest and straight line (line segment) exceeds  $max\_d$ , the data point group is segmented. The column "Correct extraction (horizontal)" shows the number of times of succeed in plane segmentation/extraction by the line fitting in which a total of 10 times of line fitting is done for each line-directional data train, the column "Correct extraction (vertical)" shows whether the extraction has successfully be done for each column-directional data train. Also, the "Nos. 1 to 5" show the conditions for plane fitting by the conventional line fitting method in which the aforementioned zig-zag shape discrimination is not adopted, and the "No. 6" shows the conditions

for the plane fitting adopted in this variant and adopting the zig-zag shape discrimination.

FIGS. 55C and 55D show the results of the plane segmentation/extraction by line fitting. FIG. 55C shows the result of the plane segmentation/extraction done according to the variant, and FIG. 55D shows the results of plane segmentation/extraction by the conventional line fitting (as comparative example). As seen in FIG. 55B, in case the plane segmentation/extraction is done by the conventional line fitting, if the threshold parameter  $max\_d$  is increased in value ( $max\_d = 25$  or  $30$ ) for the purpose of estimation, the rate of detection will be lower. If the threshold parameter  $max\_d$  is decreased in value ( $max\_d = 10$  or  $15$ ), the detection rate will be higher. Thus it will be understood that according to the present invention, however, the introduction of the zig-zag shape discrimination permits to provide excellent results of detection even if the threshold parameter is increased in value ( $max\_d = 30$ ).

More particularly, as the threshold parameter  $max\_d$  is set larger, the noise influence will be lower but the line fitting will be difficult. On the contrary, when the threshold parameter  $max\_d$  is set smaller, the line fitting will be erroneous more frequently under the influence of noises. FIG. 56A schematically illustrates an image of a floor surface, and FIGS. 56B and 56C show lines fitted by the line fitting method adopted in this variant and by the conventional line fitting method, respectively, on the basis of horizontal and vertical distance data point trains,

respectively, included in the three-dimensional distance data acquired by taking a picture of the floor surface shown in FIG. 56A. The left half of each of FIGS. 56B and 56C shows an example of the line fitted based on the line-directional pixel train (distance data train) and right half shows an example of the line fitted based on the column-directional pixel train (distance data train). As will be known from FIG. 56B, when the threshold parameter  $max\_d$  is set smaller, the noise influence will be increased. Lines cannot well be detected from a far location where the noise influence is large. On the other hand, as seen in FIG. 56C, in case the zig-zag shape discrimination is added to the conventional line fitting, even if the threshold parameter  $max\_d$  is set larger, lines can be detected even from a far location where the influence of noises is large.

Thus, plane segmentation/extraction can be done by acquiring three-dimensional distance data from an image of different stairs as above. As shown in FIGS. 11 and 12, for example, all treads can be detected as planes in any case. It should be noted that a part of the floor can be detected as another plane as shown in FIG. 12B.

According to this variant, for the plane segmentation/extraction by the line fitting method, a large threshold is initially set to segment a line. Then, the zig-zag shape discrimination is effected to segment a straight line having not any larger number of data points than the threshold but having a zig-zag shape taking the line as a line formed from a plurality of planes. Therefore, the plurality of



planes can accurately be detected based on distance information or data including noises.

Since even a small step can be detected with a high accuracy as above, stairs or the like can be recognized in an environment in which the robot apparatus can move, for example, the robot apparatus, which is of a bipedal walking type, can climb the stairs on the basis of such results of detection.

Further, any irregular floor surface formed from a plurality of planes will not erroneously be recognized as a plane on which the robot apparatus can walk and the robot apparatus will be able to move more easily.

In the foregoing, the present invention has been described in detail concerning certain preferred embodiment and variant thereof as examples with reference to the accompanying drawings. However, it should be understood by those ordinarily skilled in the art that the present invention is not limited to the embodiments but can be modified in various manners, constructed alternatively or embodied in various other forms without departing from the scope and spirit thereof as set forth and defined in the appended claims. Any arbitrary one or more of the aforementioned plane segmentation/extraction, stair recognition and stair climbing may be implemented by a hardware or a computer program is to be executed by a CPU. In the operations are to be performed under the control of the computer program, the latter may be distributed in a recording medium or via a transmission medium such as the Internet or the like.

## CLAIMS

1. A robot apparatus enabled by a locomotive means to move, the apparatus comprising:

a plane segmenting/extracting means for detecting a plane or a plurality of planes included in an environment surrounding the apparatus from three-dimensional distance data and outputting the detected plane or planes as plane information;

a stair recognizing means for recognizing, based on the plane information, a stair having a plane on which the apparatus can move and outputting stair information including information on the tread and riser of the stair; and

a stair-climb controlling means for controlling the stair climbing of the apparatus by judging, based on the stair information, whether the apparatus can climb the stair and having the apparatus autonomously position itself in relation to the stair tread when the stair is determined to be climbable.

2. The apparatus according to claim 1, further comprising a distance measuring means for acquiring the three-dimensional distance data.

3. The apparatus according to claim 1, further including legs as the locomotion means, on which the apparatus can move.

4. The apparatus according to claim 1, wherein the stair recognizing means includes:

a stair extracting means for detecting, based on the supplied plane

information, a stair having a plane on which the apparatus can move and outputting stair information not yet merged; and

a stair merging means for statistically processing a plurality of yet-to-be-merged stair information supplied from the stair extracting means and which are temporally different from each other and outputting merged stair information.

5. The apparatus according to claim 4, wherein:

the stair extracting means recognizes the dimensions and spatial positions of a tread on the basis of the plane information and outputs the resultant tread information as the yet-to-be-merged stair information; and

the stair merging means merges, when a group of two or more treads overlapping each other in a range larger than a predetermined threshold and whose difference in height from each other is smaller than a predetermined threshold has been detected from temporally continuous tread information, the tread information together into a single piece of tread information including tread information on all the treads in the tread group.

6. The apparatus according to claim 1, wherein the stair recognizing means recognizes the dimensions and spatial position of a tread on the basis of the plane information, and takes them as the tread information.

7. The apparatus according to claim 6, wherein the tread information includes information on a front edge as the outermost edge, and information on a back edge

as the innermost edge, of the tread in at least the moving direction.

8. The apparatus according to claim 7, wherein the tread information includes right- and left-margin information indicative of marginal regions adjacent to the right and left sides, respectively, of a safety zone defined by the front and back edges and in which it is estimated that it is highly probable that the apparatus can move.

9. The apparatus according to claim 7, wherein the tread information includes reference-point information indicative of the center of gravity of a region which is estimated, based on the plane information, to be a tread.

10. The apparatus according to claim 9, wherein the tread information includes position information on a reference point which is the center of gravity of a safety zone indicating defined by the front and back edges, the center of gravity of a tread region adjacent to the safety zone and opposite sides of the latter, which is a marginal region in which it is estimated that it is highly probable that the apparatus can move or a central point determined based on a group of points forming a plane which is a tread.

11. The apparatus according to claim 7, wherein the tread information includes three-dimensional coordinate information on a group of points forming together a plane which is to be a tread.

12. The apparatus according to claim 1, wherein the stair recognizing means extracts the boundary of a plane on the basis of the plane information to calculate a

polygon, and calculate the tread information from the polygon.

13. The apparatus according to claim 12, wherein the polygon is a convex polygonal region circumscribing the plane boundary extracted based on the plane information.

14. The apparatus according to claim 12, wherein the polygon is a convex polygonal region inscribing the plane boundary extracted on the basis of the plane information.

15. The apparatus according to claim 12, wherein the polygon is a non-convex polygonal region resulted from smoothing of the boundary of the plane extracted on the basis of the plane information.

16. The apparatus according to claim 1, wherein the plane information includes more than one selected for each plane from among a normal vector, plane-boundary information, center-of-gravity information, dimensions information and flatness.

17. The apparatus according to claim 3, wherein the stair-climb controlling means controls the apparatus to climb the stair after the apparatus has moved to a predetermined position opposite to the back edge of the tread on which the apparatus is moving.

18. The apparatus according to claim 17, wherein the stair-climb controlling means controls, when the back edge of the tread on which the apparatus is moving cannot be recognized, the apparatus to climb the stair after having moved to a predetermined position opposite to the front edge of a next tread to which the

apparatus is to move.

19. The apparatus according to claim 3, wherein the stair-climb controlling means controls the apparatus to climb the stair by having the apparatus make a series of operations from detection of a next tread to which the apparatus is to move up to movement to a predetermined position opposite to the next tread.
20. The apparatus according to claim 19, wherein the stair-climb controlling means searches, when a next tread or a tread after next cannot be detected from a position where the apparatus currently stands, the next tread on the basis of the previously acquired stair information.
21. The apparatus according to claim 3, wherein the stair-climb controlling means controls the apparatus to move a next tread by having the apparatus move to a predetermined position opposite to the back edge of the present tread on which the apparatus is moving, then detect the next tread and move to a predetermined position opposite to the front edge of the next tread.
22. The apparatus according to claim 3, wherein the stair-climb controlling means controls the apparatus to climb stairs on the basis of parameters specifying a position of the locomotion means in relation to a tread.
23. The apparatus according to claim 22, the parameters are determined on the basis of a height of foot lifting or lowering of the leg.
24. The apparatus according to claim 22, further comprising a parameter selecting means for selecting parameter values for an intended stair-ascending or



-descending motion.

25. The apparatus according to claim 1, further comprising:

a line fitting means for fitting lines for each group of distance data points estimated to be in one plane in a three-dimensional space; and

a planar region growing means for extracting a plurality of lines estimated to be in one plane from the group of lines fitted by the line fitting means and calculating a plane from the extracted plurality of lines,

the line fitting means can fit lines adaptively to the distribution of the distance data points.

26. The apparatus according to claim 25, wherein the line fitting means extracts a group of distance data points estimated to be in one plane on the basis of the distance between the distance data points, and re-estimates, based on the distribution of the distance data points in the distance data point group, whether the distance data point group exists in one plane.

27. The apparatus according to claim 25, wherein the line fitting means extracts lines from a group of distance data points estimated to be in the one plane, takes, as a point of interest, a distance data point whose distance to the lines is largest in the group of distance data points, judges, when the distance is smaller than a predetermined threshold, whether the distance data points in the distance data point group are unevenly distributed, and segments the distance data point group by the point of interest when the distribution is uneven.

28. The apparatus according to claim 25, wherein the line fitting means extracts a first line from the group of distance data points estimated to lie in the one plane, takes a distance data point in the group, whose distance from the first line is longest, as a point of interest, extracts a second line from the distance data point group when the distance is smaller than a predetermined threshold, judges whether a larger number of distance data points than a predetermined number exist continuously at one side of the second line, and segments the distance data point group by the point of interest when the larger number of distance data points than the predetermined number exist continuously.

29. The apparatus according to claim 25, wherein the planar region grouping means selects more than one line estimated to be in one plane and calculates a reference plane, searches lines estimated to be in the plane in which the reference plane lies as grouping lines from the group of lines, updates the reference plane with the grouping lines and repeats the grouping of the region of the reference plane, and outputs the updated plane as an updated one.

30. The apparatus according to claim 29, further comprising a plane recalculating means for recalculating a plane from the group of distance data points except for ones whose distance from the updated plane is larger than a predetermined threshold, if any, existing in the distance data point group in the updated plane.

31. The apparatus according to claim 25, wherein the planar region grouping

means estimates, based on a difference between a lines-determined plane and reference plane, whether the lines lie coplanar with the reference plane.

32. A method of controlling the motion of a robot apparatus enabled by a locomotive means to move, the method comprising the steps of:

detecting a plane or a plurality of planes included in an environment surrounding the apparatus from three-dimensional distance data and outputting the detected plane or planes as plane information;

recognizing, based on the plane information, a stair having a plane on which the apparatus can move and outputting stair information including information on the tread and riser of the stair; and

controlling the stair climbing of the apparatus by judging, based on the stair information, whether the apparatus can climb the stair and having the apparatus autonomously position itself in relation to the stair tread when the stair is determined to be climbable.

33. The method according to claim 32, further comprising the step of acquiring the three-dimensional distance data.

34. The method according to claim 32, wherein the locomotion means is legs of the robot apparatus.

35. The method according to claim 32, wherein the stair recognizing step includes:

the stair extracting step of detecting, based on the supplied plane information,

a stair having a plane on which the apparatus can move and outputting stair information not yet merged; and

the stair merging step of statistically processing a plurality of yet-to-be-merged stair information supplied from the stair extracting step and which are temporally different from each other and outputting merged stair information.

36. The method according to claim 35, wherein:

in the stair extracting step, the dimensions and spatial positions of a tread are recognized on the basis of the plane information and the resultant tread information is outputted as the yet-to-be-merged stair information; and

in the stair merging step, when a group of two or more treads overlapping each other in a range larger than a predetermined threshold and whose difference in height from each other is smaller than a predetermined threshold has been detected from temporally continuous tread information, the tread information is merged together into a single piece of tread information including tread information on all the treads in the tread group.

37. The method according to claim 32, wherein in the stair recognizing step, the dimensions and spatial position of a tread are recognized on the basis of the plane information, and takes them as the tread information.

38. The method according to claim 37, wherein the tread information includes information on a front edge as the outermost edge, and information on a back edge

as the innermost edge, of the tread in at least the moving direction.

39. The method according to claim 32, wherein in the stair recognizing step, the boundary of a plane is extracted on the basis of the plane information to calculate a polygon and the tread information is calculated from the polygon.

40. The method according to claim 34, wherein in the stair-climb controlling step, the apparatus is controlled to climb the stair after having moved to a predetermined position opposite to the back edge of the tread on which the apparatus is moving.

41. The method according to claim 40, wherein the stair-climb controlling step, when the back edge of the tread on which the apparatus is moving cannot be recognized, the apparatus is controlled to climb the stair after having moved to a predetermined position opposite to the front edge of a next tread to which the apparatus is to move.

42. The method according to claim 34, wherein in the stair-climb controlling step, the apparatus is controlled to climb the stair by making a series of operations from detection of a next tread to which the apparatus is to move up to movement to a predetermined position opposite to the next tread.

43. The method according to claim 42, wherein in the stair-climb controlling step, when a next tread or a tread after next cannot be detected from a position where the apparatus currently stands, the next tread is searched on the basis of the previously acquired stair information.

44. The method according to claim 34, wherein in the stair-climb controlling step, the apparatus is controlled to move a next tread by having the apparatus move to a predetermined position opposite to the back edge of the present tread on which the apparatus is moving, then detect the next tread and move to a predetermined position opposite to the front edge of the next tread.

45. The method according to claim 34, wherein in the stair-climb controlling step, the apparatus is controlled to climb stairs on the basis of parameters specifying a position of the locomotion means in relation to a tread.

46. The method according to claim 45, further comprising a parameter selecting step of selecting parameter values for an intended stair-ascending or -descending motion.

47. A locomotion mechanism movable by locomotion means, the mechanism comprising:

a plane segmenting/extracting means for detecting a plane or a plurality of planes included in an environment surrounding the apparatus from three-dimensional distance data and outputting the detected plane or planes as plane information;

a stair recognizing means for recognizing, based on the plane information, a stair having a plane on which the apparatus can move and outputting stair information including information on the tread and riser of the stair; and

a stair-climb controlling means for controlling the stair climbing of the



apparatus by judging, based on the stair information, whether the apparatus can climb the stair and having the apparatus autonomously position itself in relation to the stair tread when the stair is determined to be climbable.

## ABSTRACT

In a robot apparatus, a stereo vision system 1 views the environment around the robot apparatus and outputs stereo data D1 as a range image which is three-dimensional distance information calculated based on the parallax between both the eyes, a plane extraction segmentation/extractor unit 2 detects planes from the range image to recognize a plurality of planes existing in the environment (plane data D2), a stair recognition unit 3 extracts a plane the robot apparatus can climb from the plane data D2, recognizes a stair from the plane and outputs stair data D4, and a stair climber 4 outputs a climb control command D5 to implement the stair climbing on the basis of the stair data D4. Namely, a locomotion body such as the robot apparatus can autonomously climb stairs by acquiring information on the stairs.

(19) 世界知的所有権機関  
国際事務局



(43) 国際公開日  
2005 年 9 月 22 日 (22.09.2005)

PCT

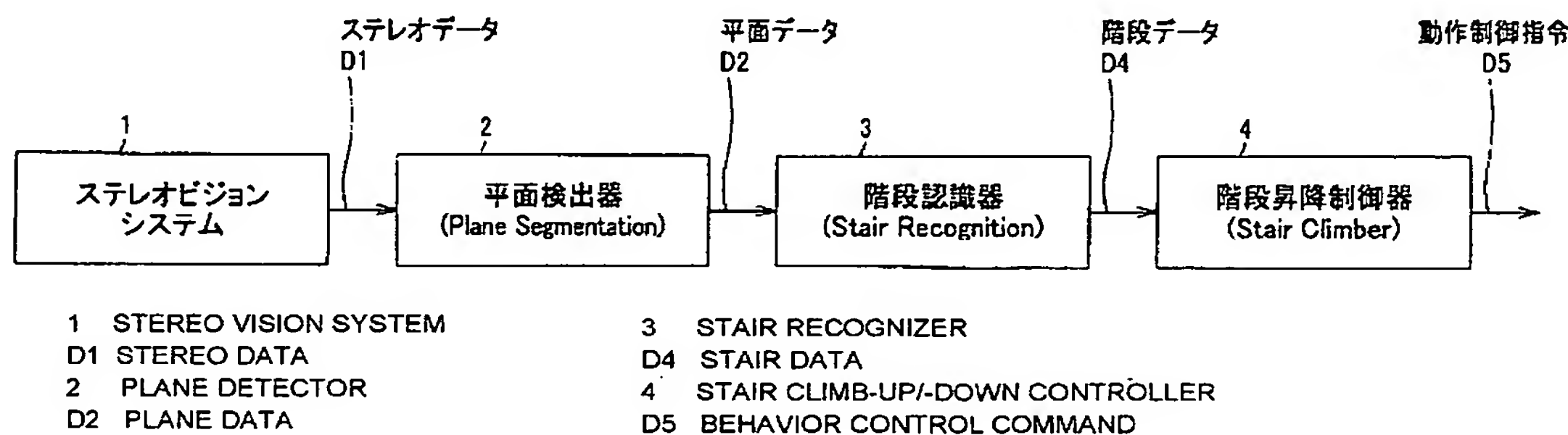
(10) 国際公開番号  
WO 2005/087452 A1

- (51) 国際特許分類<sup>7</sup>: B25J 13/00, 5/00 (74) 代理人: 小池 晃, 外(KOIKE, Akira et al.); 〒1000011 東京都千代田区内幸町一丁目 1 番 7 号 大和生命ビル 1 1 階 Tokyo (JP).
- (21) 国際出願番号: PCT/JP2005/004838
- (22) 国際出願日: 2005 年 3 月 17 日 (17.03.2005)
- (25) 国際出願の言語: 日本語
- (26) 国際公開の言語: 日本語
- (30) 優先権データ:  
特願2004-077214 2004 年 3 月 17 日 (17.03.2004) JP
- (71) 出願人 (米国を除く全ての指定国について): ソニー株式会社 (SONY CORPORATION) [JP/JP]; 〒1410001 東京都品川区北品川 6 丁目 7 番 3 5 号 Tokyo (JP).
- (72) 発明者; および
- (75) 発明者/出願人 (米国についてのみ): ゲットマン ステフェン (GUTMANN, Steffen) [DE/JP]; 〒1410001 東京都品川区北品川 6 丁目 7 番 3 5 号 ソニー株式会社内 Tokyo (JP). 福地 正樹 (FUKUCHI, Masaki) [JP/JP]; 〒1410001 東京都品川区北品川 6 丁目 7 番 3 5 号 ソニー株式会社内 Tokyo (JP).
- (81) 指定国 (表示のない限り、全ての種類の国内保護が可能): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.
- (84) 指定国 (表示のない限り、全ての種類の広域保護が可能): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), ヨーロッパ (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

[続葉有]

(54) Title: ROBOT DEVICE, BEHAVIOR CONTROL METHOD FOR THE ROBOT DEVICE, AND MOVING DEVICE

(54) 発明の名称: ロボット装置、及びその動作制御方法、並びに移動装置



(57) Abstract: A robot device observes the external world by a stereo vision system (1) and outputs stereo data (D1) as a distance image, the stereo data (D1) being three-dimensional distance information calculated by a parallax of both eyes. A plane detector (2) detects planes from the distance image to recognize the planes present in the environment. From plane data (D2), a stair recognizer (3) extracts a plane which the robot can climb up and down, recognizes a stair from that plane, and outputs stair data (D4). A stair climb-up/-down controller (4) outputs a behavior control command (D5) for realizing the movement for climbing up and down the stair by using the stair data (D4). This enables a mobile body itself to obtain information on a stair to autonomously perform the movement of stair climb up and down.

(57) 要約: ロボット装置は、ステレオビジョンシステム (1) によって外界を観測し、両眼の視差によって算出される 3 次元距離情報であるステレオデータ (D1) を距離画像として出力する。平面検出器 (2) は、この距離画像から平面を検出することで、環境内に存在する複数の平面を認識する。階段認識器 (3) は、平面データ (D2) からロボット装置が昇降可能な平面を抽出し、その平面から階段を認識し階段データ (D4) を出力する。階段昇降制御器 (4) は、この階段データ (D4) を用いて階段の昇降動作を実現する動作制御指令 (D5) を出力する。これによって、移動体自身が階段に係る情報を取得して自律的に階段昇降動作を可能とする。



添付公開書類:

- 国際調査報告書
- 請求の範囲の補正の期限前の公開であり、補正書受理の際には再公開される。

2文字コード及び他の略語については、定期発行される各PCTガゼットの巻頭に掲載されている「コードと略語のガイダンスノート」を参照。

## 明 細 書

ロボット装置、及びその動作制御方法、並びに移動装置

### 技術分野

[0001] 本発明は、例えば脚部などの移動手段を有し、複数段からなる階段の昇降動作等を可能とするロボット装置、移動装置、及びその階段昇降方法に関する。

本出願は、日本国において2004年3月17日に出願された日本特許出願番号2004-077214を基礎として優先権を主張するものであり、これらの出願は参照することにより、本出願に援用される。

### 背景技術

[0002] 従来、車輪等の移動手段を有する移動装置を含むロボット装置に対し、段差のある環境や、階段の昇降動作をさせることを可能とする技術が複数開示されている。

例えば、対象物体の形状特徴点を認識するため対象物体のノードデータを有した、図1に示す既知の階段情報を用いて昇降動作を行う方法(特許第3176701号公報)や、図2に示すように、多数の接触センサ602が保護膜603で被覆されつつマトリクス状に足平601の裏面の全面的に貼り付けられた足平を利用して、階段昇降動作を行う方法がある(特許第3278467号公報)。更に、図3に示すように、足底の脇に赤外センサを装備し、階段にランドマークテープを付与することで、階段昇降動作を行う方法がある(特許第3330710号公報)。これは、図3Aに示すように、2足歩行可能なロボット装置の足平622R/Lの左右の脇部分に複数の光センサ検出部682を設け、黒色ペイントなどの光をよく吸収する塗料で描いた直線からなる所定幅の面領域であるランドマーカ680を利用することで、対になっているセンサ出力を比較することでランドマーカ680に対する相対的な方角を検出するものである。このような足平622R/Lとランドマーカ680とを使用することで、図3Bに示すように、階段690の位置を認識することが可能となる。

しかしながら、特許第3176701号公報に記載の技術においては、既知の階段情報に基づいているため、未知の環境においては適応することができなく、例えば自律型のロボット装置などに適用することが困難である。また、特許第3278467号公報及

び特許第3330710号公報に記載の技術においては、足裏もしくは足底の脇に設けられた複数のセンサを利用して階段昇降動作を行うため、着床するまで階段の情報が取得できない。したがって、遠くからの観測が不可能である。このため、ある程度目の前に階段があることが予測されるときにしか利用することができない。

## 発明の開示

### 発明が解決しようとする課題

[0003] 本発明は、このような従来の実情に鑑みて提案されたものであり、移動体自身が階段に係る情報を取得して自律的に階段昇降動作を可能とするロボット装置及び移動装置、並びにロボット装置の動作制御方法を提供することを目的とする。

上述した目的を達成するために、本発明に係るロボット装置は、移動手段により移動可能なロボット装置において、3次元の距離データから環境内に含まれる1又は複数の平面を検出し、平面情報として出力する平面検出手段と、上記平面情報から移動可能な平面を有する階段を認識し、該階段の踏面に関する踏面情報及び蹴り上げ情報を有する階段情報を出力する階段認識手段と、上記階段情報に基づき、階段昇降可能か否かを判断し、昇降動作が可能であると判断した場合には、その踏面に対して自律的に位置決めして階段昇降動作を制御する階段昇降制御手段を有することを特徴とする。

本発明においては、移動手段として例えば脚部などを備えて移動可能なロボット装置において、階段の踏面の例えば大きさや位置などに関する踏面情報から、その足底がその踏面に載せることができる大きさか否かを判断したり、階段の段差を示す蹴り上げの情報からその高さの踏面への移動が可能か否かを判断し、移動可能であると判断した場合には自律的に位置決めすることで階段を登ったり降ったりすることが可能となる。

また、上記3次元の距離データを取得する距離計測手段を有することができ、階段を検出した際には自律的に昇降動作が可能となる。

また、上記階段認識手段は、与えられた平面情報から移動可能な平面を有する階段を検出して統合前階段情報を出力する階段検出手段と、上記階段検出手段から出力される時間的に異なる複数の統合前階段情報を統計的に処理することにより統



合した統合済階段情報を上記階段情報として出力する階段統合手段とを有することができ、例えば視野が狭いロボット装置であったり、1度の検出処理ではうまく階段認識できないような場合であっても、時間的に統計的に統合した統合済階段情報とすることで、正確かつ高域な認識結果を得ることができる。

更に、上記階段検出手段は、上記平面情報に基づき踏面の大きさ及び空間的な位置を認識し、この認識結果である踏面情報を上記統合前階段情報として出力し、上記階段統合手段は、時間的に前後する踏面情報から、所定の閾値より大きい重複領域を有しかつ相対的な高さの違いが所定の閾値以下である2以上の踏面からなる踏面群を検出した場合、当該踏面群を何れをも含む一の踏面となるよう統合することができ、統合の際には統合すべきとして選択された踏面を全て含むように統合することで、広い範囲にわたって認識結果を得ることができる。

更にまた、上記階段認識手段は、上記平面情報に基づき踏面の大きさ及び空間的な位置を認識し上記踏面情報とすることができ、踏面情報は、少なくとも移動方向に対して該踏面の手前側の境界を示すフロントエッジ及び奥側の境界を示すバックエッジの情報を含むものとしてすることができ、踏面のフロントエッジ及びバックエッジを認識するため、例えばスパイラル形状の階段などであっても正確に踏面を認識して階段昇降動作を可能にする。

また、上記フロントエッジ及びバックエッジに挟まれた領域である安全領域の左右両側に隣接した領域であって移動可能である確率が高いと推定されるマージン領域を示す右側マージン情報及び左側マージン情報、上記平面情報に基づき踏面と推定された領域の重心を示す参照点情報、踏面となる平面を構成する点群の3次元座標情報などを有することができる。これらの踏面情報を使用することで、階段昇降動作をより正確に制御することができる。

更に、上記階段認識手段は、上記平面情報に基づき平面の境界を抽出して多角形を算出し、該多角形に基づき上記踏面情報を算出することができ、例えば、視野が狭い場合、3次元距離データの信頼性が高いような場合には、上記多角形は、上記平面情報に基づき抽出された平面の境界に外接する凸多角形領域とすることができ、実際に検出されている平面を含んだ領域とすることができ。一方、ノイズが多い

距離データなどの場合には、上記多角形は、上記平面情報に基づき抽出された平面の境界に内接する凸多角形領域とすることができ、実際に検出されている平面に内包される領域とすることで、ノイズ部分をカットして正確に踏面を検出することができる。

更にまた、現在移動中の移動面におけるバックエッジに対峙した所定位置に移動した後、昇降動作を実行するよう制御することができ、例えば蹴り上げが小さい階段など、フロントエッジとバックエッジが重なるような場合には、バックエッジを目標に移動して昇降動作することができる。同様に、フロントエッジを目標に移動して昇降動作を行ってもよい。

また、上記階段昇降制御手段は、現在移動中の移動面におけるバックエッジが確認できない場合は、次に昇降動作の対象となる次段の踏面におけるフロントエッジに対峙した所定位置に移動した後、昇降動作を実行するよう制御することができ、例えば床面を移動していて階段を検出した場合、床面のバックエッジが階段初段のフロントエッジに重ならない場合があり、そのような場合は階段の初段、すなわち昇降動作の対象となる次段の踏面のフロントエッジを目標に移動して昇降動作をすることができる。

また、上記階段昇降制御手段は、次に移動対象となる踏面を検出し、当該移動対象となる踏面に対峙した所定位置に移動する一連の動作を行って昇降動作を実行するよう制御することができ、新たな踏面に移動する毎に、踏面に対してサーチ・アライン・アプローチ動作を実行することで昇降動作を可能にする。

更に、上記階段昇降制御手段は、現在位置から次に移動対象となる次段又は次段以降の踏面が検出できない場合、過去に取得した階段情報から当該移動対象となる次段の踏面を検索することができ、予め数段上又は下の階段情報を取得しておくことで、ロボット装置が自身の直近の情報が得られないような構成で視野が狭い場合であっても昇降動作を可能とする。

また、上記階段昇降制御手段は、現在の移動面におけるバックエッジに対峙した所定位置に移動した後、次の移動対象となる踏面を検出し、当該踏面におけるフロントエッジに対峙した所定位置に移動し、当該踏面に移動する昇降動作を実行するよう

制御することができ、フロントエッジ及びバックエッジを使用することで、両エッジが平行していない螺旋状の階段であっても昇降動作を可能とする。

更に、上記昇降制御手段は、踏面に対する上記移動手段の位置を規定したパラメータを使用して昇降動作を制御することができ、このパラメータは、例えば上記脚部の足上げ高さ又は足下げ高さに基づき決定されることができる。そして、階段を登る動作と降りる動作とで上記パラメータの数値を変更するパラメータ切り替え手段を有することができ、階段を登る動作であっても、降りる動作であってもパラメータ変更するのみで同様に制御することができる。

また、上記平面検出手段は、3次元空間で同一平面上にあると推定される距離データ点群毎に線分を抽出する線分抽出手段と、上記線分抽出手段によって抽出された線分群から同一平面に属すると推定される複数の線分を抽出し該複数の線分から平面を算出する平面領域拡張手段とを有し、

上記線分抽出手段は、距離データ点の分布に応じて適応的に線分を抽出することができ、線分抽出手段は、3次元の距離データが同一平面上にある場合同一直線上に並ぶことを利用して線分を抽出するが、この際、ノイズなどの影響により距離データ点の分布に違いが生じるため、この距離データの分布に応じて適応的に線分を抽出する(Adaptive Line Fitting)ことにより、ノイズに対してロバストに、精確な線分抽出を可能とし、抽出された多数の線分から線分拡張法により平面を求めるため、ノイズの影響などにより、本来複数平面が存在するのに1つの平面としたり、1つの平面しか存在しないのに複数平面としたりすることなく精確に平面抽出することができる。

更に、上記線分抽出手段は、上記距離データ点間の距離に基づき同一平面上にあると推定される距離データ点群を抽出し、該距離データ点群における距離データ点の分布に基づき、当該距離データ点群が同一平面上にあるか否かを再度推定することができ、距離データ点の3次元空間における距離に基づき一旦距離データ点群を抽出しておき、データ点の分布に基づき再度同一平面上にあるか否かを推定することにより精確に線分抽出することができる。

更にまた、上記線分抽出手段は、上記同一平面上にあると推定される距離データ点群から第1の線分を抽出し、該距離データ点群のうち該第1の線分との距離が最も

大きい距離データ点を着目点とし、当該距離が所定の閾値以下である場合に該距離データ点群から第2の線分を抽出し、該第2の線分の一方側に距離データ点が所定の数以上連続して存在するか否かを判定し、所定の数以上連続して存在する場合に該距離データ点群を該着目点にて分割することができ、例えば抽出したデータ点群の端点を結ぶ線分を第1の線分とし、上記距離が大きい点が存在する場合には、例えば最小二乗法により第2の線分を生成し、この第2の線分において一方側に連続して複数のデータ点が存在する場合には、データ点群は例えば線分に対してジグザグな形などをとっていることが想定でき、したがって抽出したデータ点群には偏りがあると判断して、上記着目点などにてデータ点群を分割することができる。

また、上記平面領域拡張手段は、同一の平面に属すると推定される1以上の線分を選択して基準平面を算出し、該基準平面と同一平面に属すると推定される線分を該線分群から拡張用線分として検索し、該拡張用線分により該基準平面を更新すると共に該基準平面の領域を拡張する処理を繰り返し、更新が終了した平面を更新済平面として出力することができ、同一平面に属するとされる線分により平面領域拡張処理及び平面更新処理を行うことができる。

更に、上記更新済平面に属する距離データ点群において、当該更新済平面との距離が所定の閾値を超える距離データ点が存在する場合、これを除いた距離データ点群から再度平面を算出する平面再算出手段を更に有することができ、更新済平面はそれに属する全線分の平均した平面として得られているため、これから大きく外れた距離データ点を除いたデータ点群から再度平面を求めることで、よりノイズなどの影響を低減した検出結果を得ることができる。

更にまた、上記平面領域拡張手段は、線分により定まる平面と上記基準平面との誤差に基づき当該線分が該基準平面と同一平面に属するか否かを推定することができ、例えば平面方程式の2乗平均誤差などに基づきノイズの影響であるのか、異なる平面なのかを判別して更に正確に平面検出することができる。

本発明に係るロボット装置の動作制御方法は、移動手段により移動可能なロボット装置の動作制御方法において、3次元の距離データから環境内に含まれる1又は複数の平面を検出し、平面情報として出力する平面検出工程と、上記平面情報から移



動可能な平面を有する階段を認識し、該階段の踏面に関する踏面情報及び蹴り上げ情報を有する階段情報を出力する階段認識工程と、上記階段情報に基づき、階段昇降可能か否かを判断し、昇降動作が可能であると判断した場合には、その踏面に対して自律的に位置決めして階段昇降動作を制御する階段昇降制御工程とを有することを特徴とする。

本発明に係る移動装置は、移動手段により移動可能な移動装置において、3次元の距離データから環境内に含まれる1又は複数の平面を検出し、平面情報として出力する平面検出手段と、上記平面情報から移動可能な平面を有する階段を認識し、該階段の踏面に関する踏面情報及び蹴り上げ情報を有する階段情報を出力する階段認識手段と、上記階段情報に基づき、階段昇降可能か否かを判断し、昇降動作が可能であると判断した場合には、その踏面に対して自律的に位置決めして階段昇降動作を制御する階段昇降制御手段を有することを特徴とする。

本発明によれば、移動手段として例えば脚部などを備えて移動可能なロボット装置及び移動装置において、階段の踏面の例えば大きさや位置などに関する踏面情報から、その足底がその踏面に載せることができる大きさか否かを判断したり、階段の段差を示す蹴り上げの情報からその高さの踏面への移動が可能か否かを判断し、移動可能であると判断した場合には自律的に位置決めすることで階段を登ったり降ったりすることが可能となる。

本発明の更に他の目的、本発明によって得られる利点は、以下において図面を参照して説明される実施形態から一層明らかにされるであろう。

#### 図面の簡単な説明

- [0004] [図1]図1は、従来の昇降動作を説明する図である。
- [図2]図2は、従来の昇降動作を説明する図である。
- [図3]図3A、図3Bは、従来の昇降動作を説明する図である。
- [図4]図4は、本発明の実施の形態におけるロボット装置の概観を示す斜視図である。
- 。
- [図5]図5は、ロボット装置が具備する関節自由度構成を模式的に示す図である。
- [図6]図6は、ロボット装置の制御システム構成を示す模式図である。

[図7]図7は、ロボット装置がステレオデータから階段昇降動作を発現するまでの処理を実行するシステムを示す機能ブロック図である。

[図8]図8Aは、ロボット装置が外界を撮影している様子を示す模式図、図8Bは、ロボット装置の足底の大きさを示す図である。

[図9]図9は、階段検出を説明する図であって、図9Aは、階段を正面から見た図、図9Bは、階段を側面から見た図、図9Cは、階段を斜めから見た図である。

[図10]図10は、階段検出の他の例を示す説明する図であって、図10Aは、階段を正面から見た図、図10Bは、階段を側面から見た図、図10Cは、階段を斜めから見た図である。

[図11]図11は、図9の階段を検出した結果の一例を示す図であって、図11Aは、図9の階段を撮影した場合の画像を示す模式図、図11B乃至図11Dは、図11Aに示す画像から取得した3次元の距離データを示す図である。

[図12]図12は、図10の階段を検出した結果の一例を示す図であって、図12Aは、図10の階段を撮影した場合の画像を示す模式図、図12B乃至図12Dは、図12Aに示す画像から取得した3次元の距離データを示す図である。

[図13]図13Aは、階段を撮影した画像を示す模式図、図13Bは、図13Aから取得した3次元距離データから4つの平面領域A、B、C、Dを検出した結果を示す図である。階段を検出した結果の一例を示す図である。

[図14]図14は、階段認識器を示す機能ブロック図である。

[図15]図15は、階段検出処理の手順を示すフローチャートである。

[図16]図16A、図16Bは、多角形を示す模式図である。

[図17]図17は、Melkmanのアルゴリズムを説明するための模式図である。

[図18]図18A、図18Bは、Sklanskyのアルゴリズムにより多角形を求める方法を説明するための模式図である。

[図19]図19は、非凸多角形形状の階段について発生する問題を説明するための模式図であって、図19Aは、入力される平面を示す図、図19Bは、凸包による非凸多角形形状の階段の多角形表現結果を示す図である。

[図20]図20は、平滑化によって入力平面を包含する多角形を求める方法を示す模



式図であって、図20Aは、入力された平面を示す図、図20Bは、入力平面を示す多角形から不連続なギャップを除去し平滑化した多角形を示す図、図20Cは、図20Bで得られた多角形に対してラインフィッティングにより更に平滑化した多角形を示す図である。

[図21]図21は、ギャップ除去とラインフィットによる平滑化によって入力平面を包含する多角形を求める処理のプログラム例を示す図である。

[図22]図22A、図22Bは、階段パラメータの算出方法を説明するための模式図である。

[図23]図23は、最終的に認識される踏面及び階段パラメータを説明するための模式図である。

[図24]図24A、図24Bは、階段を示す模式図である。

[図25]図25は、階段統合処理の方法を示すフローチャートである。

[図26]図26は、オーバーラップしている階段データを統合する処理を説明するための模式図である。

[図27]図27は、アライン動作を説明するための図である。

[図28]図28は、アプローチ動作を説明するための模式図である。

[図29]図29は、階段昇降動作の手順を示すフローチャートである。

[図30]図30は、サーチ・アライン・アプローチ処理方法を示すフローチャートである。

[図31]図31は、昇降動作処理の方法を示すフローチャートである。

[図32]図32は、ロボット装置が認識しているか又は認識する予定の階段面を示す模式図である。

[図33]図33は、昇降動作処理の方法を示すフローチャートである。

[図34]図34は、昇降動作処理の方法を示すフローチャートである。

[図35]図35Aは、ロボット装置により認識されている踏面と足底の関係を説明するための図であり、図35Bは、各部の寸法を示す図である。

[図36]図36は、ロボット装置が昇降動作を行った様子を撮影したものをトレースした図である。

[図37]図37は、ロボット装置が昇降動作を行った様子を撮影したものをトレースした

図である。

[図38]図38は、単一の段部とロボット装置の足底の関係を示す図である。

[図39]図39は、単一の凹部とロボット装置の足底の関係を示す図である。

[図40]図40は、本変形例における平面検出装置を示す機能ブロック図である。

[図41]図41は、テクスチャを付与する手段を有しているロボット装置を説明するための図である。

[図42]図42は、本変形例における線分拡張法による平面検出方法を説明する図である。

[図43]図43は、線分拡張法による平面検出処理を示すフローチャートである。

[図44]図44は、本変形例における線分抽出部における処理の詳細を示すフローチャートである。

[図45]図45は、距離データ点の分布の様子を示す図であって、図45Aは、データの分布が線分に対してジグザグ形である場合、図45Bは、ノイズなどにより線分近傍に一樣に分布している場合を示す模式図である。

[図46]図46は、本変形例におけるZig-Zag-Shape判別方法を示すフローチャートである。

[図47]図47は、上記Zig-Zag-Shape判別処理のプログラム例を示す図である。

[図48]図48は、Zig-Zag-Shape判別処理を行う処理部を示すブロック図である。

[図49]図49は、本変形例における領域拡張処理を説明するための模式図である。

[図50]図50は、本変形例における領域拡張部における領域種を検索する処理及び領域拡張処理の手順を示すフローチャートである。

[図51]図51は、端点と直線との距離が等しくても平面方程式の2乗平均誤差rmsが異なる例を示す図であって、図51Aは、ノイズなどの影響により線分が平面からずれている場合、図51Bは、線分が属する他の平面が存在する場合を示す模式図である。

[図52]図52は、領域種の選択処理を示す図である。

[図53]図53は、領域拡張処理を示す図である。

[図54]図54Aは、ロボット装置が立った状態で床面を見下ろした際の床面を示す模

式図、図54Bは、縦軸をx、横軸をy、各データ点の濃淡でz軸を表現して3次元距離データ及び、行方向の画素列から線分抽出処理にて同一平面に存在するとされるデータ点群から直線を検出したものを示す図、図54Cは、図54Bに示す直線群から領域拡張処理により得られた平面領域を示す図である。

【図55】図55は、床面に段差を一段置いたときの本変形例における平面検出方法と従来の平面検出方法との結果の違いを説明するための図であって、図55Aは、観察された画像を示す模式図、図55Bは、実験条件を示す図、図55Cは、本変形例における平面検出方法により平面検出された結果を示す図、図55Dは、従来の平面検出方法により平面検出された結果を示す図である。

【図56】図56Aは、床面を撮影した画像を示す模式図、図56B及び図56Cは、図56Aに示す床面を撮影して取得した3次元距離データから水平方向及び垂直方向の距離データ点列から、それぞれ本変形例の線分検出により検出した線分及び従来の線分検出により検出した線分を示す図である。

発明を実施するための最良の形態

以下、本発明を適用した具体的な実施の形態について、図面を参照しながら詳細に説明する。この実施の形態は、本発明を、周囲の環境に存在する階段などの段差を認識する段差認識装置を搭載した自律的に動作可能なロボット装置に適用したものである。

本実施の形態におけるロボット装置は、ステレオビジョンなどにより得られた距離情報（距離データ）から抽出した複数平面から階段を認識し、この階段認識結果を利用して階段昇降動作を可能とするものである。

#### (1) ロボット装置

ここでは、まず、このようなロボット装置の一例として2足歩行タイプのロボット装置を例にとって説明する。このロボット装置は、住環境その他の日常生活上の様々な場面における人的活動を支援する実用ロボットであり、内部状態（怒り、悲しみ、喜び、楽しみ等）に応じて行動できるほか、人間が行う基本的な動作を表出できるエンターテインメントロボット装置である。なお、ここでは、2足歩行型のロボット装置を例にとって説明するが、階段認識装置は、2足歩行のロボット装置に限らず、脚式移動型のロボット装置に搭載すればロボット装置に階段昇降動作を実行させることができる。

図4は、本実施の形態におけるロボット装置の概観を示す斜視図である。図4に示すように、ロボット装置201は、体幹部ユニット202の所定の位置に頭部ユニット203が連結されると共に、左右2つの腕部ユニット204R/Lと、左右2つの脚部ユニット205R/Lが連結されて構成されている(ただし、R及びLの各々は、右及び左の各々を示す接尾辞である。以下において同じ。)

このロボット装置201が具備する関節自由度構成を図5に模式的に示す。頭部ユニット203を支持する首関節は、首関節ヨー軸101と、首関節ピッチ軸102と、首関節ロール軸103という3自由度を有している。

また、上肢を構成する各々の腕部ユニット204R/Lは、肩関節ピッチ軸107と、肩関節ロール軸108と、上腕ヨー軸109と、肘関節ピッチ軸110と、前腕ヨー軸111と、手首関節ピッチ軸112と、手首関節ロール軸113と、手部114とで構成される。手部114は、実際には、複数本の指を含む多関節・多自由度構造体である。ただし、手部114の動作は、ロボット装置201の姿勢制御や歩行制御に対する寄与や影響が少ないので、本明細書では簡単のため、ゼロ自由度と仮定する。したがって、各腕部は7自由度を有するとする。

また、体幹部ユニット202は、体幹ピッチ軸104と、体幹ロール軸105と、体幹ヨー軸106という3自由度を有する。

また、下肢を構成する各々の脚部ユニット205R/Lは、股関節ヨー軸115と、股関節ピッチ軸116と、股関節ロール軸117と、膝関節ピッチ軸118と、足首関節ピッチ軸119と、足首関節ロール軸120と、足底121とで構成される。本明細書中では、股関節ピッチ軸116と股関節ロール軸117の交点は、ロボット装置201の股関節位置を定義する。人体の足底121は、実際には多関節・多自由度の足底を含んだ構造体であるが、本明細書においては、簡単のためロボット装置201の足底は、ゼロ自由度とする。したがって、各脚部は、6自由度で構成される。

以上を総括すれば、ロボット装置201全体としては、合計で $3 + 7 \times 2 + 3 + 6 \times 2 = 32$ 自由度を有することになる。ただし、エンターテインメント向けのロボット装置201が必ずしも32自由度に限定されるわけではない。設計・制作上の制約条件や要求仕様等に応じて、自由度すなわち関節数を適宜増減することができることはいうまでも

ない。

上述したようなロボット装置201がもつ各自由度は、実際にはアクチュエータを用いて実装される。外観上で余分な膨らみを排してヒトの自然体形状に近似させること、2足歩行という不安定構造体に対して姿勢制御を行うこと等の要請から、アクチュエータは小型かつ軽量であることが好ましい。

このようなロボット装置は、ロボット装置全体の動作を制御する制御システムを例えば体幹部ユニット202等に備える。図6は、ロボット装置201の制御システム構成を示す模式図である。図6に示すように、制御システムは、ユーザ入力等に動的に反応して情緒判断や感情表現を司る思考制御モジュール200と、アクチュエータ350の駆動等、ロボット装置201の全身協調運動を制御する運動制御モジュール300とで構成される。

思考制御モジュール200は、情緒判断や感情表現に関する演算処理を実行するCPU (Central Processing Unit) 211や、RAM (Random Access Memory) 212、ROM (Read Only Memory) 213及び外部記憶装置(ハード・ディスク・ドライブ等) 214等で構成され、モジュール内で自己完結した処理を行うことができる、独立駆動型の情報処理装置である。

この思考制御モジュール200は、画像入力装置251から入力される画像データや音声入力装置252から入力される音声データ等、外界からの刺激等に従って、ロボット装置201の現在の感情や意思を決定する。すなわち、上述したように、入力される画像データからユーザの表情を認識し、その情報をロボット装置201の感情や意思に反映させることで、ユーザの表情に応じた行動を発現することができる。ここで、画像入力装置251は、例えばCCD (Charge Coupled Device) カメラを複数備えており、これらのカメラにより撮像した画像から距離画像を得ることができる。また、音声入力装置252は、例えばマイクロホンを複数備えている。

思考制御モジュール200は、意思決定に基づいた動作又は行動シーケンス、すなわち四肢の運動を実行するように、運動制御モジュール300に対して指令を発行する。

一方の運動制御モジュール300は、ロボット装置201の全身協調運動を制御する



CPU311や、RAM312、ROM313及び外部記憶装置(ハード・ディスク・ドライブ等)314等で構成され、モジュール内で自己完結した処理を行うことができる独立駆動型の情報処理装置である。また、外部記憶装置314には、例えば、オフラインで算出された歩行パターンや目標とするZMP軌道、その他の行動計画を蓄積することができる。

この運動制御モジュール300には、図5に示したロボット装置201の全身に分散するそれぞれの関節自由度を実現するアクチュエータ350、対象物との距離を測定する距離計測センサ(図示せず)、体幹部ユニット202の姿勢や傾斜を計測する姿勢センサ351、左右の足底の離床又は着床を検出する接地確認センサ352、353、足底121の足底121に設けられる荷重センサ、バッテリー等の電源を管理する電源制御装置354等の各種の装置が、バス・インターフェース(I/F)310経由で接続されている。ここで、姿勢センサ351は、例えば加速度センサとジャイロ・センサの組み合わせによって構成され、接地確認センサ352、353は、近接センサ又はマイクロ・スイッチ等で構成される。

思考制御モジュール200と運動制御モジュール300は、共通のプラットフォーム上で構築され、両者間はバス・インターフェース210、310を介して相互接続されている。

運動制御モジュール300では、思考制御モジュール200から指示された行動を体現すべく、各アクチュエータ350による全身協調運動を制御する。すなわち、CPU311は、思考制御モジュール200から指示された行動に応じた動作パターンを外部記憶装置314から取り出し、又は、内部的に動作パターンを生成する。そして、CPU311は、指定された動作パターンに従って、足部運動、ZMP軌道、体幹運動、上肢運動、腰部水平位置及び高さ等を設定するとともに、これらの設定内容に従った動作を指示する指令値を各アクチュエータ350に転送する。

また、CPU311は、姿勢センサ351の出力信号によりロボット装置201の体幹部ユニット202の姿勢や傾きを検出するとともに、各接地確認センサ352、353の出力信号により各脚部ユニット205R/Lが遊脚又は立脚の何れの状態であるかを検出することによって、ロボット装置201の全身協調運動を適応的に制御することができる。更



に、CPU311は、ZMP位置が常にZMP安定領域の中心に向かうように、ロボット装置201の姿勢や動作を制御する。

また、運動制御モジュール300は、思考制御モジュール200において決定された意思通りの行動がどの程度発現されたか、すなわち処理の状況を、思考制御モジュール200に返すようになっている。このようにしてロボット装置201は、制御プログラムに基づいて自己及び周囲の状況を判断し、自律的に行動することができる。

## (2) ロボット装置の動作制御方法

上述のロボット装置においては、頭部ユニット203にステレオビジョンシステムを搭載し、外界の3次元距離情報を取得することができる。次に、このようなロボット装置などに好適に搭載されるものであって、ロボット装置が、ステレオビジョンシステムにより周囲の環境から獲得した3次元距離データを使用して平面を検出し、この平面検出結果に基づき階段を認識し、この階段認識結果を使用して階段昇降動作を行う一連の処理について説明する。

図7は、ロボット装置がステレオデータから階段昇降動作を発現するまでの処理を実行するシステムを示す機能ブロック図である。図7に示すように、ロボット装置は、3次元の距離データを取得する距離データ計測手段としてのステレオビジョンシステム(Stereo Vision System) 1と、ステレオビジョンシステム1からステレオデータD1が入力され、このステレオデータD1から環境内の平面を検出する平面検出器(Plane Segmentation/Extractor) 2と、平面検出器2から出力される平面データD2から階段を認識する階段認識器(Stair Recognition) 3と、階段認識部2により認識された認識結果である階段データD4を使用して階段昇降動作をするための動作制御指令D5を出力する階段昇降制御器(Stair Climber) 4とを備える。

そして、ロボット装置は、先ず、ステレオビジョンによって外界を観測し、両眼の視差によって算出される3次元距離情報であるステレオデータD1を画像として出力する。すなわち、人間の両眼に相当する左右2つのカメラからの画像入力を各画素近傍毎に比較し、その視差から対象までの距離を推定し、3次元距離情報を画像として出力(距離画像)する。この距離画像から平面検出器2によって平面を検出することで、環境内に存在する複数の平面を認識することができる。更に、階段認識器3によって、

これら平面からロボット装置が昇降可能な平面を抽出し、その平面から階段を認識し階段データD4を出力する。そして階段昇降制御器4が階段データD4を用いて階段の昇降動作を実現する動作制御指令D5を出力する。

図8Aは、ロボット装置201が外界を撮影している様子を示す模式図である。床面をx-y平面とし、高さ方向をz方向としたとき、図8Aに示すように、画像入力部(ステレオカメラ)を頭部ユニット203に有するロボット装置201の視野範囲は、ロボット装置201の前方の所定範囲となる。

ロボット装置201は、上述したCPU211において、画像入力装置251からのカラー画像及び視差画像と、各アクチュエータ350の全ての関節角度等のセンサデータなどが入力されて各種の処理を実行するソフトウェア構成を実現する。

本実施の形態のロボット装置201におけるソフトウェアは、オブジェクト単位で構成され、ロボット装置の位置、移動量、周囲の障害物、及び環境地図等を認識し、ロボット装置が最終的に取るべき行動についての行動列を出力する各種認識処理等を行うことができる。なお、ロボット装置の位置を示す座標として、例えば、ランドマーク等の特定の物体等に基づく所定位置を座標の原点としたワールド基準系のカメラ座標系(以下、絶対座標ともいう。)と、ロボット装置自身を中心(座標の原点)としたロボット中心座標系(以下、相対座標ともいう。)との2つの座標を使用する。

ステレオビジョンシステム1では、カラー画像及びステレオカメラによる視差画像などの画像データが撮像された時間において、センサデータから割り出した関節角を使用してロボット装置201が中心に固定されたロボット中心座標系を頭部ユニット203に設けられた画像入力装置251の座標系へ変換する。この場合、本実施の形態においては、ロボット中心座標系からカメラ座標系の同次変換行列等を導出し、この同次変換行列とこれに対応する3次元距離データからなる距離画像を出力する。

本実施の形態におけるロボット装置は、自身の視野内に含まれる階段を認識することができ、その認識結果(以下、階段データという。)を使用して階段昇降動作を可能とする。したがって、階段昇降動作のためには、ロボット装置は、階段の大きさが自身の足底の大きさより小さいか、階段の高さが登ることができる又は降りることができる高さか否かなどの、階段の大きさについて様々な判断を行う必要がある。

ここで、本実施の形態においては、ロボット装置の足底の大きさを図8Bとした場合について説明する。すなわち、図8Bに示すように、ロボット装置201の前進方向をx軸方向、床面に平行でx方向と直交する方法をy方向としたとし、ロボット装置201が直立した際の両足のy方向の幅をfeet base width足底の大きさであて足首(脚部と足底の接続部)から前側の部分を足底前幅foot\_front\_size、足首から後ろ側の部分を足底の足底後ろ幅foot\_back\_sizeとするものとする。

ロボット装置201が環境内から検出する階段としては、例えば図9、図10に示すようなものがある。図9A、図10Aは、階段を正面から見た図、図9B、図10Bは、階段を側面から見た図、図9C、図10Cは、階段を斜めから見た図である。

ここで、人間、ロボット装置などが階段を昇降するために使用する面(足又は可動脚部を載せる面)を踏面といい、一の踏面からその次の踏面までの高さ(階段1段の階段の高さ)を蹴り上げという。また、以下では、階段は、地面に近い方から登るに従って1段目、2段目とカウントすることとする。

図9に示す階段ST1は、段数が3段の階段であり、蹴り上げ4cm、1, 2段面の踏面の大きさは幅30cm、奥行き10cm、最上段である3段目の踏面のみ、幅30cm、奥行き21cmとなっている。また、図10に示す階段ST2も、段数が3段の階段であり、蹴り上げ3cm、1, 2段面の踏面の大きさは幅33cm、奥行き12cm、最上段である3段目の踏面のみ、幅33cm、奥行き32cmとなっている。ロボット装置がこれらの階段を認識した結果は後述する。

平面検出器2は、ステレオビジョン等の距離計測器から出力される距離情報(ステレオデータD1)から環境内に存在する複数の平面を検出し、平面データD2を出力する。平面の検出方法としては後述する線分拡張法その他、ハフ変換を利用した公知の平面検出技術を適用することができる。ただし、ノイズを含む距離データから階段のように複数平面を検出するには、後述する線分拡張法などによる平面検出を行うと正確に平面を検出することができる。

図11、図12は、階段を検出した結果の一例を示す図である。図11及び図12は、後述する平面検出方法により、それぞれ図9及び図10に示す階段を撮影した画像から3次元距離データを取得して平面検出した例である。すなわち、図11Aは、図9

の階段を撮影した場合の画像を示す模式図、図11B乃至図11Dは、図11Aに示す画像から取得した3次元の距離データを示す図である。また、図12Aは、図10の階段を撮影した場合の画像を示す模式図、図12B乃至図12Dは、図12Aに示す画像から取得した3次元の距離データを示す図である。図11及び図12に示すように、何れの場合も全ての踏面を平面として検出できている。図11Bは、下から1段目、2段目、3段目の踏面が平面検出されている例を示す。また、図12Bは、床面の一部も他の平面として検出成功していることを示す。

すなわち、図13Aに示すように、例えば階段ST2を撮影した距離画像から平面検出すると、図13Bに示すように、領域A〜Dがそれぞれ、床面、1段目、2段目、3段目の踏面を示す平面として検出される。各領域A〜Dに含まれる同一領域に示す点群は、それぞれ同一平面を構成すると推定された距離データ点群を示している。

階段認識器3には、こうして平面検出器2が検出した平面データD2が入力され、階段の形状、すなわち踏面の大きさ、階段の高さ(蹴り上げの大きさ)などを認識する。ここで、本実施の形態における階段認識器3は、詳細は後述するが、ロボット装置201が認識した踏面に含まれる領域(多角形)の手前側(ロボット装置に距離的に近い側)の境界(Front Edge)(以下、フロントエッジFEという。 )と、踏面の奥側(ロボット装置に距離的に遠い側)の境界(Back Edge)(以下、バックエッジBEという。 )とを階段データとして認識する。そして、階段昇降制御器4が階段データを利用して階段昇降動作を制御する。

次に、ロボット装置の階段昇降制御方法について具体的に説明する。なお、以下では、第1に、ロボット装置の階段認識方法、第2に認識した階段を利用して行う階段昇降動作、最後に平面検出方法の具体例として線分拡張法による平面検出方法の順にて説明する。

図14は、図7に示す階段認識器を示す機能ブロック図である。図14に示すように、階段認識器3は、平面検出器2から出力された平面データD2から階段を検出する階段検出器(Stair Extraction)5と、この階段検出器5が検出した階段データD3の時系列データ、すなわち異なる時間に検出された複数の階段データD3を統合することで更に正確に階段を認識する処理を行う階段統合器(Stair Merging)6とを有し、階段



統合器にて統合した階段データD4が階段認識器3の出力となる。

階段検出器5は、平面検出器2から入力される平面データD2から階段を検出するが、

平面検出器2から入力される平面データD2は、1つの平面につき以下に示す複数の情報を有し、ステレオビジョンシステム1によって取り込まれた画像から検出された複数の平面毎の平面データが入力される。

すなわち、平面データD2は、平面毎に

1-1: 平面を構成する点の数(number of supporting point)

1-2: 平面の中心となる点

1-3: 平面パラメータ(法線ベクトル、原点からの距離)

1-4: 平面を構成する多角形の境界

から構成される情報を有する。

この平面データに基づき、ロボット装置は、自身が接地している床面や踏面などの接地面と略水平な平面を選択し、下記の情報(以下、階段パラメータという。)を算出する。すなわち、

2-1: フロントエッジFE、バックエッジBE

2-2: 階段の高さ

である。

ロボット装置が認識するフロントエッジFE、バックエッジBEとは、上述した如く、階段の踏面の境界(線)を示すものであって、ロボット装置が対峙した場合に、多角形においてロボット装置に近い側の境界(手前側の境界)をフロントエッジFE、ロボット装置から距離が離れている側の境界(奥側の境界)をバックエッジBEとする。これは、後述するように、例えば平面を構成する点を全て含む最小の多角形を求め、その手前側又は奥側の境界とすることができる。フロントエッジFE及びバックエッジBEの情報としては、これらの端点の情報などとすることができる。また、上記多角形から階段の幅W(width)、階段の長さ(length)などの情報を得ることができる。また、階段の高さ(蹴り上げ)は、与えられた平面データD2の平面の中心点を利用して、2の平面の中心点間の高さ差としたり、上記多角形を求めた際の重心点を利用して2の重心点の

高さの差としたりすることができる。なお、蹴り上げは、前段のバックエッジBEと後段のフロントエッジFEとの高さの差としたりしてもよい。

また、本実施の形態においては、フロントエッジFE及びバックエッジBEに加えて更にフロントエッジFE及びバックエッジBEに挟まれた領域(安全領域)の左右に隣接する領域であって移動可能である確率が高いと推定される領域をマージン(領域)として認識する。これらの求め方については後述する。このマージンを求めることにより、移動可能であると推定する踏面の領域を広く認識することができる。更に、踏面を構成するデータ点群の数や、上述の重心点などを1つ規定した参照点の情報などの情報(階段パラメータ)のセットを階段データD3とすることができる。

以上の階段データから下記の条件を満たす平面(階段)を抽出する。

3-1: フロントエッジFE及びバックエッジBEの長さが所定の閾値以上

3-2: 階段の高さ(height)が所定の閾値以下

また、その他、

3-3: 階段の幅W(width)が所定の閾値以上

3-4: 階段の長さL(length)が所定の閾値以上

など同時に満たすものを抽出することが好ましい。

図15は、階段検出器5の階段検出処理の手順を示すフローチャートである。図15に示すように、先ず、入力された平面データの平面パラメータなどに基づき、入力された平面が例えば接地面と水平であるか否かなど、歩行又は移動可能である平面か否かを判断する(ステップS1)。ここで、どのような平面が水平であるか又は移動可能であるかの条件は、ロボット装置の機能に応じて設定すればよい。例えば、入力平面の平面ベクトルを $n(n_x, n_y, n_z)$ とした場合、 $|\sin_{-1} n_z| > \min_{th}$ であれば水平であると判断することができる。ここで、 $\min_{th}$ は、水平面を判断するための閾値であり、例えば、使用する距離データ、平面検出など精度を考慮して $\min_{th} = 80^\circ$  などとして、水平面に対して $\pm 10^\circ$  までの傾きであれば水平と判断して検出するなどとすることができる。または、例えば、 $\pm 30^\circ$  程度の傾きがあっても歩行または移動可能であれば、それらの角度範囲の平面を抽出するようにすればよい。ステップS1にて水平でないと判断した場合(ステップS1: No)、検出に失敗したことを出力して処理を終了し、次



の平面データについての処理を実行する。

次に、平面が水平である場合(ステップS1:Yes)、平面の境界(形状)を認識するための処理を行う。ここでは、例えばSklanskyのアルゴリズム(J. Sklansky, "Measuring concavity on a rectangular mosaic", IEEE Trans Comput.21,1974,pp.1355-1364)やMelkmanのアルゴリズム(Melkman A., "On-line Construction of the Convex Hull of a Simple Polygon" Information Processing Letters 25,1987,p.11)などの凸包や、ノイズ除去による平滑化によって、入力平面を包含する多角形を求める(ステップS2)。そして、この多角形の前後の境界線をフロントエッジ及びバックエッジなどの階段パラメータを求める(ステップS3)。そして、フロントエッジ及びバックエッジの両境界線から、本実施の形態においては、階段踏面を示す平面における幅W(width)及び長さL(length)を求め、これらの値が所定の閾値より大きいかどうか判断する(ステップS4)。踏面の幅及び長さが所定の閾値以上でない場合(ステップS4:No)、ロボット装置が移動可能な平面ではないとし、次の平面データについて再びステップS1からの処理を繰り返す。

平面の幅及び長さが所定の閾値以上である場合(ステップS4:No)、移動可能な踏面であると判断し左右のマージン(Left Margin, Right Margin)を計算し(ステップS5)、これらの情報を階段データD3として出力する。

次に、ステップS2において凸包によって入力平面を包含する多角形を求める方法について説明する。図16は、凸多角形を示す模式図であって、図16Aは、入力された一の平面に属すると判断されたサポーティングポイント全て(同一平面であって連続した領域に含まれるとされる距離データ点群)が含まれる領域を示し、図16Bは、図16Aに示す図形から求めた凸多角形である。ここで示す凸多角形は、与えられた平面図形(サポーティングポイントが含まれる領域)を含む最小の凸集合を求める凸包(convex hull)を利用したものとすることができる。なお、Gで示す点は、後述するが、踏面の幅Wを求める際に使用するもので、例えばサポーティングポイントが含まれる領域の重心などの点(参照点)を示す。

このような凸包を利用して凸多角形を求めるアルゴリズムの例として、上述したように、Melkmanのアルゴリズム、Sklanskyのアルゴリズムなどがある。図17は、Melkmanの

アルゴリズムを説明するための模式図である。図17に示すように、与えられた図形に含まれる点を3点P1, P2, P3抽出し、点P1, P2を結ぶ線分を引き、点P1, P3、, P2, P3を通る直線を引く。これにより、3点P1, P2, P3からなる三角形AR4を含む5つの領域AR1〜AR5に区画される。そして、次に選択した点P4がどの領域に含まれるかを判断して多角形を形成しなおすという処理を繰り返して凸多角形を更新していく。例えばP4が、領域AR1に存在する場合、P1, P2, P4, P3の順序で結んだ線分に囲まれた領域が更新された凸多角形となる。また、領域AR3, AR4に点P4が存在する場合には、それぞれP1, P2, P3, P4の順序で結んだ線分に囲まれた領域、P1, P4, P2, P3の順序で結んだ線分に囲まれた領域として凸多角形を更新する。一方、領域AR4、すなわち凸多角形内部に点Pが含まれる場合は、凸多角形は更新せず、また、領域AR2に点P4がある場合は、点P3は除きP1, P2, P3の順序で結んだ線分に囲まれた領域として凸多角形を更新する。本実施の形態においては、全てのサポーティングポイントに対して、各点に含まれる領域を考慮して凸多角形を生成することができる。

図18は、Sklanskyのアルゴリズムにより多角形を求める方法を説明するための模式図である。Sklanskyのアルゴリズムにより抽出される多角形は、Weakly Externally Visible Polygonと呼ばれるものであるが、上述のSklanskyのアル比して計算量が少なく、したがって高速演算が可能である。

与えられた図形を包含する凸多角形を求める場合、図18Aに示すように、与えられた図形131の境界上の任意の点xから図形131を含む円132に対して半直線を引く。このとき、点xから円132に引いた半直線のうち、図形131を横切らない半直線を引くことができた場合、この点は凸多角形の境界を構成する点であるとする。一方、図18Bに示すように、与えられた図形133の境界上任意の他の点yから図形133を含む円134に半直線を引いた場合には、図形133を横切らない半直線を引くことができない。この場合は、この他の点yは凸多角形の境界を構成しないものとする。以上のようにして、順次各点が凸多角形の境界を構成するか否かを判断して選択された点のみからの図形を求めると、図16Aに示すような図形が得られる。

この図形を包含する凸多角形を求めることで、図16Bの凸多角形を得ることができ

る。ここで、本実施の形態においては、ステレオビジョンシステム1の精度、特性などを考慮し、図16Aから凸多角形を求める場合には、図16Bに示すように、図16Aの図形に外接する凸多角形を求めるものとして説明するが、カメラの精度、特性などを考慮し、例えば図16Aの図形に内接する凸多角形を求めるようにしてもよいことは勿論である。また、これらの方法を平面の傾き度合いや、周囲の状況に応じて使い分けるようにしてもよい。

また、図15のステップS2において凸包によって入力平面を包含する多角形を求めた場合、非凸多角形形状の階段について問題が発生する。図19は、この問題を示す模式図であって、図19Aは、入力される平面でありstep0は非凸多角形形状の階段である。図19Bは、凸包によるstep0の多角形表現結果であり、非凸部分に関して望ましい結果と大きな乖離が生じている。このような非凸多角形を扱う方法として、ギャップ除去とラインフィットによる平滑化によって多角形を求める方法が考えられる。

ギャップ除去とラインフィットによる平滑化によって入力平面を包含する多角形を求める方法について説明する。図20は、平滑化を示す模式図であって、図20Aは、入力された一の平面に属すると判断されたサポーティングポイント全て(同一平面であって連続した領域に含まれるとされる距離データ点群)が含まれる領域である入力多角形(input polygon)を示し、図20Bは、入力平面を示す多角形から不連続なギャップを除去し(close gaps)平滑化した多角形(ギャップ除去多角形:gaps closed polygon)を示し、図20Cは、図20Bで得られた多角形に対してラインフィッティングにより(fit line segments)更に平滑化した多角形(平滑化多角形:smoothed polygon)である。ここで、図21は、ギャップ除去とラインフィットによる平滑化によって入力平面を包含する多角形を求める処理のプログラム例を示す図であり、多角形から不連続なギャップを除去するClose gaps処理、及び得られた多角形に対してラインフィッティングにより更に平滑化するFit line segments処理を示している。

まず、ギャップ除去の方法について説明する。多角形を示す頂点から連続する3つの頂点を選び、この中央の点が端点を結んだ直線から大きく離れている場合にこの中央の点を除去する。残った頂点について、この処理を除去する点が無くなるまで続ける。

次に、ラインフィットの方法について説明する。多角形を示す頂点から連続する3つの頂点を選び、最小二乗法によりこれらの3点を近似する直線とこの直線と3点の誤差を求める。求められた全ての近似直線と誤差について、誤差の小さい順に並べ、誤差がある閾値より小さい場合に、中央の点を除去し、端点の位置を近似直線によって再計算する。この処理を除去する点が無くなるまで続ける。

次に、図15のステップS3における処理について説明する。まず、得られた多角形(図16B、図20C)から上述した階段パラメータを算出する。図22は、階段パラメータの算出方法を説明するための模式図である。図22Aに示すように、得られた多角形140が、点141〜点147により囲まれる領域であるものとする。ここで、ロボット装置201からみて多角形140の前側の境界を構成する線分がフロントエッジFE、奥側の境界を構成する線分がバックエッジBEである。

階段踏面の幅Wは、フロントエッジFEの中心点 $C_{FE}$ と参照点Gとを結ぶ線分の長さをd1、バックエッジBEの中心点 $C_{BE}$ と参照点Gとを結ぶ線分の長さをd2としたとき、幅 $W = d1 + d2$ とすることができる。

ここで、参照点Gは、踏面となる平面の略中心であればよく、例えば、全サポーティングポイントの中心点としたり、多角形140の重心としたり、フロントエッジFE及びバックエッジBEの端点を結んだ図22Bに示す安全領域152の重心としたりすることができる。

こうして得られたフロントエッジFE、バックエッジBE、参照点Gに基づき、階段の長さL及び幅Wを求めステップS4の処理を実行する。階段の長さLは、フロントエッジFE及びバックエッジBEの長さのうち例えば短い方としたり、以下に示す左右のマーヅンを含めたフロントエッジFE及び左右のマーヅンを含めたバックエッジBEのうち長い方としたりすることができる。

次に、ステップS5におけるマーヅン算出方法について説明する。図23は、最終的に認識される踏面及び階段パラメータを説明するための模式図である。本実施の形態においては、図22B、図23に示すように、安全領域152の左右端部にマーヅン $M_1$ 、 $M_2$ を設け、この左右のマーヅン $M_1$ 、 $M_2$ を含んだ領域151を最終的に踏面として認識するものとする。左右のマーヅン $M_1$ 、 $M_2$ は、フロントエッジFE及びバックエッジ



BEにて規定される安全領域152の外側に多角形がはみ出している場合、先ず、それらの点を選択する。図22Aにおいて、例えば右マージン $M_2$ を求める場合であれば、点142, 143, 146である。安全領域152の右側に隣接した右マージン $M_2$ を求める際は、安全領域152から最も離れた点である点142を選択し、この点142からフロントエッジFE、バックエッジBEに対して垂線を下ろす。そして、これら垂線及びフロントエッジFE、バックエッジBEに囲まれる領域151を踏面として認識するものとする。なお、このマージンの求め方としては、点142を通り、フロントエッジFE又はバックエッジBEと交差する線分を引くのみでもよい。

ここで、図22Bに示すように、フロントエッジFEと同一直線における左マージン $M_1$ の長さを $lmf$ とし、バックエッジBEと同一直線における左マージン $M_1$ の長さを $lbm$ とする。同様に、フロントエッジFE、バックエッジBEとそれぞれ同一直線における右マージン $M_2$ の長さを $rfm$ ,  $rbm$ とする。

このようにして多角形からフロントエッジFE、バックエッジBEを求めて階段を認識することの効果について説明しておく。図24A及び図24Bは、2種類の階段を示す模式図である。図24Aに示すのは、図9、図10に示したような踏面が長方形の形状の階段であるが、図24Bは、螺旋(spiral)状の階段である。この図24Bのような螺旋階段の場合、フロントエッジFEに対してバックエッジBEは平行でない。したがって、例えば単に検出した平面から長方形領域を抽出してしまうようなアルゴリズムを適用できない場合がある。したがって、本実施の形態のように、検出した平面から多角形を求め、フロントエッジFE及びバックエッジBEを求めることにより、このような螺旋階段であってもロボット装置が昇降動作することが可能となる。

次に、図14に示す階段統合器6について説明する。階段統合器6は、階段検出器5によって検出された階段データ(階段パラメータ)D3を入力とし、それらの階段データD3を時間的に統合することで、より正確かつ高域な階段の情報を推定するものである。例えば、ロボット装置の視野が狭い場合などにおいて、一度に階段全体を認識できない場合がある。そのような場合、例えば、例えば前フレームなどの古い階段データと例えば現フレームなどの新しい階段データとの中で、空間的にオーバーラップしている階段の組を探し、オーバーラップしている階段を統合することにより、新しい

仮想的な階段を定義する。この作業をオーバーラップする階段がなくなるまで続けることによって正確な階段を認識することが可能となる。

図25は、階段統合器6における階段統合処理の方法を示すフローチャートである。まず、現在の階段データ(New Stairs)と古い階段データ(Old Stairs)とを入力とし(ステップS11)、新しい階段データ及び古い階段データの全てを1つの集合(ujunion)とする(ステップS12)。これら合わされた階段データ集合において、空間的にオーバーラップしている階段データを検索し(ステップS13)、オーバーラップしている組がある場合には(ステップS14:Yes)、それらの階段を統合して当該階段データ集合に登録する(ステップS15)。そして、ステップS13、14の処理を空間的にオーバーラップする階段の組がなくなるまで続け(ステップS14:No)、最終的に更新された階段データ集合を階段データD4として出力する。

図26は、オーバーラップしている階段データを統合するステップS13における処理を説明するための模式図である。図26は、空間的にオーバーラップしている階段データST11, ST12を示す。空間的にオーバーラップしているかどうかの判断には、例えば2つの階段データST11, ST12の参照点Gにおける高さの差(距離)と、左右のマージンを含めた踏面の領域が重なっている面積の大きさを利用することができる。すなわち、2つの階段の重心 $G_{11}$ ,  $G_{12}$ の高さの差が閾値(maxdz)以下であり、かつ、オーバーラップしている面積の大きさが閾値(minarea)以上ある場合には、これらの2つの階段データが示す踏面はオーバーラップしていると判断することができる。その場合には図26の下図に示すように、階段データST11, ST12を統合して重心 $G_{13}$ の階段データST13とする。

ここで、統合に際しては、階段データST11, ST12を含む外枠の領域をステップST13とし、統合前の階段データST11及び階段データST12の左右のマージンを除いた安全領域を含む領域を統合後の新たな安全領域165とし、階段データST13からこの安全領域165を除いた領域をマージン $M_1$ ,  $M_2$ とする。統合後の安全領域165により統合されたフロントエッジFE及びバックエッジBEを求めることができる。

すなわち、統合後の階段データST13におけるフロントエッジFEの両端点は、階段データST11のフロントエッジFEと階段データST12のフロントエッジFEの左右の端



点を比較し、右端点163はより右側にある方、左端点はより左側にある方とされる。また、フロントエッジFEのラインの位置は、階段データST11のフロントエッジFEと階段データST12のフロントエッジFEを比較してよりロボット装置に近い方(手前側)のライン位置とされる。同様にバックエッジBEもより奥側にある方の位置が選択され、より左右に広がるようにその左右の端点161, 162が選択される。

なお、統合方法はこれに限るものではない。本実施の形態においては、ロボット装置の視界などを考慮してフロントエッジFE及びバックエッジBEにより決まる四角形の領域、統合データST13が共に最も大きくなるように統合するものとしているが、例えば、十分視界が広い場合や、距離データの精度が十分高い場合においては、2つの階段データを単に合わせた領域を統合後の階段データとするなどしてもよい。また、統合後の参照点Gは、階段データST11と階段データST12に含まれるサポーティングポイントの数の比に応じて重み付き平均をとって求めることができる。

次に、図7に示す階段昇降制御器4について説明する。階段昇降制御器4は、階段検出器及び階段統合器6によって統合されて検出された階段データD4を用いて実際にロボット装置が階段の昇降動作を行うための制御を行う。この昇降制御には、階段を探す動作も含むものとする。

本実施の形態において実現した階段昇降動作は、次の5つのステートマシンとして構築することができる。

#### 4-1:サーチ(Search)動作

ロボット装置の頭部に搭載されたステレオビジョンシステムにて距離画像を取得するため、首を振って周囲を見廻し、環境の情報を集める。

#### 4-2:アライン(Align)動作

階段に対して正対し、かつ、決められた一定量の距離位置に移動する。図27は、アライン動作を説明するための図である。図27において、領域170がロボット装置の認識した階段踏面の1段目であるとする。アライン動作においては、踏面170のフロントエッジFEの中心点から直交する方向に所定距離 $ad$  (align\_distance)離れた目的とする位置(以下、アライン位置という。)に移動する。この場合、現在のロボット装置の位置が点171であって、目的とするアライン位置が点172である場合、両者の距離

が所定の閾値 $\max\_d$ 以上離れている場合及びフロントエッジFEに直行する方向とロボット装置の向く方向との角度差が所定の閾値 $\max\_a$ 以上ある場合、ロボット装置は目的とするアライン位置172に移動を開始し、これらの条件を満たしたとき、アライン動作が完了したと判断するものとする。これらの閾値としては、例えば $\max\_d=3(\text{cm})$ 、 $\max\_a=2^\circ$  とすることができる。

#### 4-3: アプローチ (Approach) 動作

アプローチ動作では、ロボット装置が階段の直前まで移動する。図28は、アプローチ動作を説明するための模式図である。図28に示すように、階段と正対しAlign\_distanceだけ離れた目的位置であるアライン位置172に移動してアライン動作を終了したロボット装置201は、階段の昇降を行うために、踏面170のフロントエッジFEの中心点 $C_{FE}$ とロボット装置201とが正対し、かつその距離が所定の値 $ax(\text{approach\_x})$ となる目的の位置(以下、アプローチ位置という。)に移動する。

#### 4-4: クライム (Climb) 動作

階段認識によって得られた階段データを元に階段昇降動作をする。次の踏面(段)に移動した場合であって、次の段の踏面が観測されている場合には、更に昇る又は下る動作を続ける。この動作を次の段がなくなるまで続けて行うことにより階段昇降動作が実現される。

#### 4-5: フィニッシュ (Finish) 動作

クライム動作にて階段を登る動作をした場合は、最上段にいることを確認し、最上段の中央へ移動する。クライム動作にて降りる動作をした場合、又は図9に示すような階段ST1である場合には例えば向きを変えるなどして降りる動作をした場合には、床面に降りたことを認識することで階段昇降動作は終了される。

以上の階段昇降処理の動作方法について更に詳細に説明する。図29は、階段昇降動作の手順を示すフローチャートである。図29に示すように、階段昇降制御の処理が開始されると、サーチ (Search)・アライン (Align)・アプローチ (Approach) 動作によって、階段を検索し、検索した階段に対して対峙した所定位置に移動(アライン)し、階段の1段目に近づくアプローチ動作を実行する(ステップS21)。このサーチ・アライン・アプローチ動作が成功した場合(ステップS22: Yes)、後述する方法にて階段

昇降を行い(ステップS23)、成功したことを出力する。近づくことに失敗した場合(ステップS22:No)、失敗したことを出力して処理を終了する。この場合は、もう一度、ステップS21の処理から繰り返すなどする。

ここで、ステップS21におけるサーチ・アライン・アプローチのシーケンスは通常ロボット装置が物体や目的地に到達するために用いる制御と同じである。具体的には例えば以下に示すような方法がある。図30は、サーチ・アライン・アプローチ処理方法を示すフローチャートである。

図30に示すように、サーチ・アライン・アプローチが開始されると、サーチ動作(1)を実行する(ステップS32)。サーチ動作(1)では、首を振りできるだけ広い範囲の情報を集める動作とする。その結果、周囲に昇降可能な階段があるかどうかを判断する(ステップS32)。ここで、検出された階段の中で1段目の踏面を構成する平面の高さ $n_z$ を利用し、高さが $step\_min\_z < n_z < step\_max\_z$ を満たす場合に、昇降可能であると判断する。昇降可能な階段があった場合(ステップS32:Yes)、階段を近くで認識しなおすために、階段に対して決められた距離( $align\_distance$ )まで移動するアライン動作を実行し(ステップS33)、昇降しようとしている階段を再度認識する(ステップS34)。このステップS34の動作がサーチ動作(2)である。

そして、昇降可能な階段であるか否かを再度確認し(ステップS35)、サーチ動作(2)が成功している場合には、再認識した階段に対して対峙しかつ所定の距離のアライン位置に移動完了できているか否か、すなわちステップS33のアライン動作が成功しているか否かを確認し(ステップS36)、昇降可能な階段があり、かつアラインされている場合には(ステップS35、S36:Yes)、初段の階段のフロントエッジまで進むアプローチ動作を実行する(ステップS37)。一方、ステップS35にて昇降可能な階段がない場合にはステップS31の処理に戻り、ステップS36にてアライン動作が成功していない場合にはステップS33の処理から繰り返す。

次に、ステップS22における階段昇降動作について説明する。階段昇降動作は、一段上又は下の段(以下、次段という。)を現在の踏面からロボット装置自身が認識できる場合の昇降動作処理1と、現在の移動面から次段は認識できない場合等であつて、2段以上、上の段又は下の段(以下、2段以上先の段という。)を認識できる場合

の昇降動作処理2と、複数段の踏面が認識可能な場合の昇降動作処理3とがある。図31、図33、図34は、それぞれ昇降動作処理1〜3の処理方法を示すフローチャートである。ここで、以下の説明においては、現在移動中の段(床面を含む)をstep-0、次段の段をstep-1、その更に次の段をstep-2、m段先の段をstep-mとする。

まず、次段(step-1)の踏面を観測・認識できる場合の昇降動作処理について説明する。図31に示すように、昇降動作処理1が開始されると、階段を登る／降りる動作(クライム動作(1))を実行する(ステップS41)。このクライム動作(1)では、上述のステップS32において、階段の高さ $n_z$ を認識できているため、この高さ $n_z$ の正負の判断をすることにより、昇降モードを切り替える。すなわち、 $n_z < 0$ であれば降りる動作であり、 $n_z > 0$ であれば登る動作となる。登る場合と降りる場合では、後述するように、クライム動作に使用する制御パラメータの値が異なる。すなわち、制御パラメータを切り替えるのみで、登る動作と降りる動作を切り替え実行することができる。

そして、クライム動作(1)が成功したか否かを判断し(ステップS42)、成功した場合(ステップS42:Yes)は、サーチ動作(3)を実行する(ステップS43)。このサーチ動作(3)は、ステレオビジョンシステムが搭載された頭部ユニットを動かし、周囲の距離データを取得して次の階段を検出する処理であり、サーチ動作(2)などとはほぼ同様の動作処理である。

そして、ステップS35と同様、サーチ動作(3)が成功したら(ステップS44:Yes)、再びステップS41の処理を繰り返す。階段が検索されなかった場合(ステップS44:No)、階段を登り切ったか、又は下りきったものと判断し、階段面の中央など所定の位置まで移動するなどのフィニッシュ動作を実行し(ステップS45)、処理を終了する。

次に、ステップS41のクライム動作(1)について詳細に説明する。図32は、ロボット装置が認識しているか又は認識する予定の階段面を示す模式図である。図32に示すように、例えば、現在移動中のロボット装置の足底121L/Rが踏面181にあるとする。また、この図32においては、フロントエッジFE及びバックエッジBEに挟まれる安全領域及びこれに隣接する左右のマージン $M_1$ 、 $M_2$ を踏面と認識することとする。このとき、ロボット装置は、次の次段(step-2)の踏面182を認識できている。ここで、踏面181と踏面182との間にはその蹴り上げなどによりギャップ184が存在するものと



する。昇降動作(1)では、現在の段(step-0)の踏面181から次の段(step-1)の踏面182に移動(クライム動作)可能であるかを判断するが、例えば、以下の基準を満たすものを移動可能と判断するものとする。

5-1:次段(step-1)の踏面182のフロントエッジ $FE_1$ に十分近く、角度のずれが所定の閾値以下

5-2:次段(step-1)の踏面182の大きさが十分に大きい

5-3:フロントエッジ $FE_1$ から足底121L/Rの後端までの距離front\_xが指定された昇降モードにおける制御パラメータfront\_x\_limitより大きい

5-4:バックエッジ $BE_1$ から足底121L/Rの前端までの距離back\_xが昇降モードにおける制御パラメータback\_x\_limitより大きい

ここで、更に次段(step-2)の踏面183を観測できた場合、次段(step-1)の踏面182の参照点における高さ $z1$ と更に次の段(step-2)の踏面183の参照点における高さ $z2$ の差( $z2-z1$ )から、次段(step-1)の踏面182からその次の段(step-2)の踏面183へのクライム動作が登りであるか下りであるかを判断することができる。なお、2段先の段(step-2)の踏面183が認識できない場合は、現在の昇降状態が維持されるものとすればよい。

上記の5-1〜5-4からクライム動作可能と判断した場合は、階段を登る又は降りるクライム動作を実行する。ロボット装置は、現在の段(step-0)の踏面181において、踏面181のバックエッジ $BE_0$ にアラインしている。ここで、踏面181と踏面182との間のギャップ184が大きい場合には、次段(step-1)の踏面182のフロントエッジ $FE_1$ にアラインする動作をした後に、次段に移動し、そのバックエッジ $BE_1$ にアラインする。そして、次のクライム動作にて、次の段(step-2)の踏面183のフロントエッジ $FE_2$ にアラインし、踏面183に移動し、そのバックエッジ $BE_2$ にアラインする。すなわち、例えば、次の段の踏面のフロントエッジFEにアラインして、昇降動作し、その移動した踏面のバックエッジBEにアラインするまでをクライム動作とする。

また、ギャップ184が小さい場合には、現在の段(step-0)の踏面181のバックエッジ $BE_0$ と次段の踏面182におけるフロントエッジ $FE_1$ がほぼ一致するものとして、何れか一方のエッジにのみアライン動作するようにしてもよい。すなわち、現在の踏面181

のバックエッジ $BE_0$ にアラインして次段の踏面182に移動し、そのバックエッジ $BE_1$ にアラインするか、次段の踏面182のフロントエッジ $FE_1$ にアラインして次段の踏面に移動し、更に次の段の踏面183のフロントエッジ $FE_1$ にアラインすればよい。この場合、クライム動作とは、次段のフロントエッジ $FE$ にアランする処理を省略し、次段に移動して移動した踏面のバックエッジ $BE$ にアライン動作を実行する処理となる。

以上の昇降動作処理1は、ロボット装置が階段を昇降動作中に次に移動可能な段(step-1)の踏面を観測できる場合に適用することができる。例えば2足歩行ロボット装置であれば、自身の足元を見下ろすことが可能なステレオビジョンシステム1を搭載しておく必要がある。

次に、次段の踏面を観測・認識可能な場合の昇降動作処理1とは異なり、ロボット装置の頭部ユニットと体幹部ユニットとの接続部の可動角の制約などにより、現在の段(step-0)における踏面から次段(step-1)の踏面を観測・認識することができず、2段先(step-2)又はそれ以上先の段(step-m)の踏面を認識可能な場合の昇降動作処理2、3について説明する。まず、2段先の段(step-2)の踏面を認識可能な場合について説明すると、図32に示すように、昇降動作処理2が開始されると、上述と同様に、サーチ動作(4)を実行する(ステップS51)。このサーチ動作(4)では、ステレオビジョンシステム1により2段先の段(step-2)の踏面を観察・認識する。このサーチ動作(4)は、2段先の段(step-2)の踏面を認識する以外、上述のステップS43などと同様の処理である。

そして、クライム動作(2)を実行する(ステップS52)。クライム動作(2)は、クライム動作(1)と同様の動作である。この場合においても、クライム動作における階段を登る動作と降りる動作の切り替えは、同じく次段(step-1)の踏面の高さ $n_z$ により判断することができる。次段(step-1)の踏面とは、現在の段(step-0)の踏面より時間的に前に移動していた段の踏面において観測された踏面である。

そして、クライム動作が成功した場合(ステップS53:Yes)、ステップS51にてサーチした結果、2段先の階段(step-2)が検出されていれば(ステップS54:Yes)、それを次の階段として更新し(step-1=step-2)(ステップS56)、ステップS51からの処理を繰り返す。ステップS51のサーチ動作(4)にて2段先の段(step-2)の踏面が検出さ



れていない場合(ステップS55:No)、フィニッシュ動作を実行し(ステップS55)、処理を終了する。

なお、ここでは2段先の段の踏面を観測できることとして説明したが、3段以降先の踏面を観測できる場合であっても同様にして処理すればよい。

次に、複数段(以下、 $m$ 段)先までの踏面が観測・認識可能な場合についての昇降動作を昇降動作3として説明する。

図34に示すように、1〜 $n$ 段が観測・認識できている場合に昇降動作を行う場合、まず、サーチ動作(5)を実行する。サーチ動作(5)は、基本的には、ステップS51と同様であるが、認識可能な $m$ 段先の段(step- $m$ )までの踏面を観測対象としている点異なる。

そして、クライム動作(3)を $k$ 段分実行する(ステップS62)。クライム動作(3)においても、現在までに観測されている複数の踏面の高さの差から昇降モードを決定することができる。すなわち、 $i$ 番目、 $i-1$ 番目の踏面の高さ $z_i - z_{i-1}$ が負であれば階段を下りる動作となり、0又は正であれば階段を登る動作モードとすればよい。このクライム動作(3)において移動する踏面の情報は、現在の段より時間的に $m$ 段前にて観測済みのデータである。

そして、昇降動作が成功した場合(ステップS63:Yes)、現在の踏面より先の階段データが観測できたか否か、すなわち $m+n > k$ であるか否かが判定され(ステップS64)、 $m+n > k$ であれば(ステップS64:Yes)、step- $(k+1)$ 〜step- $(n+m)$ を、step-1〜step- $(m+n-k)$ ( $n=m$ )として、更新し(ステップS66)、ステップS61からの処理を繰り返す。一方、 $m < 0$ であれば(ステップS64:No)、次に移動する対象となる踏面が存在しないため、フィニッシュ動作を実行し(ステップS65)、処理を終了する。

以上説明したように、階段昇降動作処理1〜3においては、クライム動作における登る動作と降りる動作とで使用する制御パラメータを変更するのみで、階段を登ることも、降りることも同様の手順にて実行することができる。階段昇降動作に使用する制御パラメータは、ロボット装置の足底の現在の踏面に対する位置を規制するためのものである。

図35Aは、ロボット装置により認識されている踏面と足底の関係を説明するための

図、図35Bは、クライム動作に使用する制御パラメータの一例を示す図である。

図35Aに示す各制御パラメータは、以下を示す。

step\_min\_z: 現在の段と次段の段との高さの差(蹴り上げ)の昇降可能最小値

step\_max\_z: 現在の段と次段の段との高さの差(蹴り上げ)の昇降可能最大値

ad (align\_distance): アライン位置におけるフロントエッジFEとロボット装置との間の距離

ax (approach\_x): アプローチ位置におけるフロントエッジFEとロボット装置との間の距離

front\_x\_limit: 踏面におけるフロントエッジFEと足底121の後端部との距離の限界値 (minimal x-value)

back\_x\_limit: 踏面におけるバックエッジBEと足底121の前端部との距離の限界値 (maximal x-value)

back\_x\_desired: バックエッジBEと足底121の前端部との距離の欲求値(desired value)

階段を登る場合には、蹴り上げがstep\_min\_zより小さい場合は段差がある(階段)と見なさないものとし、蹴り上げがstep\_max\_zより大きい場合には、階段昇降不可と判断する。同様に、階段を降りる場合には、蹴り上げがstep\_max\_zより大きい場合には、階段と見なさないものとし、step\_min\_zより大きい場合には、階段昇降不可と判断するものとする。

align\_distanceは、アライン動作をする場合にのみ使用するパラメータであり、階段を登る／降りる動作を実行する階段昇降動作処理を開始する場合、すなわち、初段の段に対する昇降動作を行う場合に使用される。同様に、approach\_xも、アプローチ動作をする場合にのみ使用するパラメータであり、階段を登る／降りる動作を実行する階段昇降動作処理を開始する場合に使用される。

front\_x\_limit及びback\_x\_limitは、ロボット装置が認識している踏面と、ロボット装置の足底の関係を規定するもので、踏面のバックエッジBE及びフロントエッジFEと、足底の端部との距離、すなわちその踏面に移動した場合における踏面のあまり部分がこれらの値より小さい場合には、その踏面に移動できない、又は移動できたとしても次

の昇降動作が不能であると判断される。ここで、登り動作において、front\_x\_limit及びback\_x\_limitの値が何れも負であることは、踏面が足底より小さいことを許すことを示す。すなわち、登り動作においては、踏面は足底より小さい場合であっても移動可能と判断することができる。

back\_x\_desiredは、現在の踏面におけるバックエッジBEに対してロボット装置がアラインしたい位置におけるバックエッジBEと足底前端部との距離を示すものであり、図35Bに示すように、通常、登る場合であれば、back\_x\_desiredは、バックエッジBEよりも手前、本実施の形態においては、バックエッジBEから15mm手前の位置となっており、一方、降りる動作であれば、バックエッジBEより足底がはみ出した位置、本実施の形態においては、5mmはみ出した位置となる。これは、登る動作であれば、次段に足上げして移動するまでにある程度の距離が必要であるのに対し、降りる動作においては、そのような距離が不要であると共に、踏面からはみ出すような位置の方が次段又はそれ以降の踏面の観測・認識が容易となるためである。

図36及び図37は、図35に示す制御パラメータを使用して実際にロボット装置が昇降動作を行った様子を撮影したものをトレースした図である。図36は、ロボット装置が階段を登る動作を示している。最上段から番号順に、ステップS31のサーチ動作(1)実行の様子(No. 1)、ステップS33のアライン動作実行の様子(No. 2)、ステップS32のサーチ動作(2)実行の様子(No. 3)、ステップS37のアプローチ動作実行の様子(No. 4)、ステップS51のサーチ動作(4)実行の様子(No. 5)、ステップS52のクライム動作(2)実行の様子(No. 6)、ステップS52のクライム動作(2)の続きであって現在の踏面のバックエッジBEにアラインしている様子(No. 7)、ステップS51のサーチ動作(4)実行の様子(No. 8)、…を示し、サーチ動作(4)をして次段の踏面が観測されなかった場合(No. 17)に、階段昇降動作終了(フィニッシュ)動作を行っている様子(No. 18)を示す。

図37は、降りる動作を示すものであり、図36の登る動作と同様に、サーチ動作(No. 1、No. 4、No. 7、No. 10、No. 13、No. 16)、クライム動作(アライン動作含む)を繰り返し(No. 5、No. 6、No. 8、No. 9、No. 11、No. 12、No. 14、No. 15)、次段の踏面が観測されなくなった時点でフィニッシュ動作を行って(No. 18)、昇降

動作を終了する。

次に、本実施の形態における変形例について説明する。以上の説明においては、階段を登る動作、降りる動作について説明したが、本実施の形態の制御方法に係るアルゴリズムを適用すれば、複数の段からなる階段のみならず、1段のみからなる段部や、1段の凹部が存在しても、その段差が所定の値以下であれば、移動を可能にするものである。

先ず、単一の段部に登る動作について説明する。図38は、単一の段部とロボット装置の足底の関係を示す図である。図38において、191で示すのは、床面( $z=0$ )からの高さが $z_1=30$ である段部を示し、ここでは、ロボット装置が次段(step1)となる段部191の紙面下側から上側に移動する場合について説明する。段部191に対し紙面下側における移動面の高さが $z_0=0$ 、紙面上側における移動面の高さが $z_2=0$ である場合であって、現在、高さ $z_0$ の移動面を移動しているとした場合、 $z_1-z_0=30>0$ であるので次段の段部191への移動は登り動作と判断し、 $z_2-z_1=-30<0$ であるので段部191から次の領域への移動は降りる動作であると判断することができる。この判断に応じてクライム動作において、上述した制御パラメータの値を変更すればよい。ここで、図35Bに示す制御パラメータを使用した場合、登る動作の場合の  $ront\_x\_limit$  及び  $back\_x\_limit$  は、何れも負の値であって、ロボット装置の足底121が図38に示すように、段部191からはみ出した状態であっても移動可能と判断することを示す。

次に、単一の窪み(凹部)に降りる動作について説明する。図39は、単一の凹部とロボット装置の足底の関係を示す図である。図39において、192で示すのは、床面( $z=0$ )からの高さが $z_1=-30$ である凹部を示し、ここでは、ロボット装置が次段(step1)となる凹部192の紙面下側から上側に移動する場合について説明する。凹部192に対し紙面下側における移動面の高さが $z_0=0$ 、紙面上側における移動面の高さが $z_2=0$ である場合であって、現在、高さ $z_0$ の移動面を移動しているとした場合、 $z_1-z_0=-30<0$ であるので次段の凹部192への移動は降りる動作と判断し、 $z_2-z_1=30>0$ であるので凹部192から次の領域への移動は登る動作であると判断することができる。したがって、段部191と同様、この判断に応じてクライム動作において、上述



した制御パラメータの値を変更すればよい。ここで、図35Bに示す制御パラメータを使用した場合、降りる動作の場合のront\_x\_limit及びback\_x\_limitは、何れも正の値であって、ロボット装置の足底121が図39に示すように、凹部191より小さい場合にのみ移動可能と判断する。

本実施の形態においては、検出された平面から例えば水平など、移動可能と判断できる平面を抽出し、その領域を含む多角形から階段の踏面を認識する。そして、多角形のフロントエッジFE及びバックエッジBEなどの踏面に関する情報と、その床面からの高さを含む階段情報を利用して階段昇降動作を行う。昇降動作においては、移動する踏面をサーチ動作を実行し、サーチできた踏面のフロントエッジFE又は現在の移動面におけるバックエッジBEに対してアライン動作を実行し、次段の移動面と現在の移動面との高さの違いから登る動作か降りる動作かを判断して制御パラメータを切り替えることにより、通常の矩形の踏面からなる階段のみならず、スパイラル形状の階段などの昇降動作が可能となると共に、登る動作も降りる動作も制御パラメータを変更するのみで同一の手順にて実行することができる。したがって、階段のみならず、単一の段部や、単一の凹部などへの移動も同一の制御方法にて移動可能となる。

また、階段認識においては、認識した階段を時間的に統合していくため、例えばロボット装置の大きさに対して階段が大きい、ロボット装置に搭載されるステレオビジョンシステムの位置の制約などの理由で視野が限られたロボット装置などにおいても、高域に亘って階段情報を認識することができる。また、この階段情報を利用して昇降動作する際、同じくステレオビジョンシステムの位置の制約などの理由で次段の踏面が観測・認識できない場合であっても、過去に観測・認識してある階段情報を利用して昇降動作を同様に行うことができる。

本変形例における平面検出装置は、線分拡張法により、視野内において支配的な平面だけでなく、例えば階段など複数の平面が存在する場合であっても確実に複数平面を検出することができ、平面を検出する際に抽出する線分抽出において、距離データの点の分布に応じて適応的に線分をフィッティングさせることにより計測ノイズに対してロバストな平面検出結果を得ることができるものである。

図40は、本変形例における平面検出装置を示す機能ブロック図である。図40に示すように、平面検出装置100は、3次元の距離データを取得する距離データ計測手段としてのステレオビジョンシステム(Stereo Vision System) 1と、3次元の距離データからなる距離画像に存在する平面を線分拡張法により検出する平面検出部2とを有する。平面検出部2は、画像を構成する距離データ点から同一平面にあると推定される距離データ点群を選択し、この距離データ点群毎に線分を抽出する線分抽出部2aと、画像内に含まれる、線分抽出部2aによって抽出された全線分からなる線分群から、該画像内に存在する1又は複数の平面領域を検出する領域拡張部2bとを有する。領域拡張部2bは、線分群から同一平面上に存在すると推定される任意の3本の線分を選択し、これらから基準平面を求める。そして、選択した3本の線分に隣接する線分がこの基準平面と同一平面に属するか否かを判定し、同一平面に属すると判定した場合にはその領域拡張用線分としての線分により基準平面を更新すると共に基準平面の領域を拡張する。

線分抽出部2aは、その距離画像における列または行毎の各データ列において、3次元空間内で同一平面上にあると推定される距離データ点群を抽出し、この距離データ点群から距離データ点群の分布に応じて1以上の線分を生成する。すなわち、分布に偏りがあると判断された場合には、データ点群は同一平面上にないと判断し、データ点群を分割し、分割したデータ点群それぞれについて再度分布に偏りがあるかを判断する処理を繰り返し、分布に偏りがない場合にはそのデータ点群から線分を生成する。全てのデータ列について以上の処理を行い、生成した線分群D11を領域拡張部2bに出力する。

領域拡張部2bは、この線分群D11において、同一の平面に属すると推定される線分を3本選択し、これらから基準平面としての種となる平面を求める。この種となる平面の領域(領域種:seed region)に対して、該領域種と同一平面に属する線分を順次統合していくことで拡張していく領域拡張によって距離画像を複数の平面に分割し、平面群D2を出力する。

ロボット装置201は、障害物回避や階段昇降など平面の情報が必要なとき、または定期的にこれらの処理を行うことによって、階段や床面、壁といった歩行に重要な平



面の情報を取得する。

ここで、このような3次元距離データをステレオビジョンシステム1によって取得するためには、階段ST2の表面に模様(テクスチャ)が必要となる。すなわち、2台のカメラによる視差により得ることができるため、模様がなければ視差が算出できず、正確に距離を計測することができない。すなわち、ステレオビジョンシステムにおける距離データの計測精度は、計測対象のテクスチャに依存することになる。なお、視差とは、空間中のある点が左目及び右目に写像される点の違いを示し、そのカメラからの距離に応じて変化するものである。

そこで、図41に示すように、ロボット装置の頭部ユニットに、ステレオビジョンシステムを構成するステレオカメラ11R/Lを備えると共に、例えば同じく頭部ユニットなどに投射手段としての例えば赤外光などを出力する光源12を設ける。この光源12は、模様がないう階段ST3、その他テクスチャがないか少ない物体、壁などの対象物に対してこれを投射(照射)し、ランダムなパターンPTを付与する模様付与手段として作用する。なお、ランダムパターンPTを形成して距離画像を取得できるものであれば、ランダムパターンPTを付与する手段は赤外光を投射する光源などには限らず、例えばロボット装置自ら対象物に模様を書いたりしてもよいが、赤外光であれば、人間の目にはみえないものの、ロボット装置に搭載されるCCDカメラなどにおいては観測可能なパターンを付与することができる。

次に、平面検出装置100の平面検出部2について説明する。この平面検出部2は、線分拡張法を使用して平面を検出するものであり、図42は、線分拡張法による平面検出方法を説明する図である。線分拡張法による平面検出では、図42に示すように、まず、焦点Fから撮影された画像11において、行方向または列方向のデータ列における処理をする。画像内の例えば行方向の画素列(image row:イメージロウ)において、距離データ点が同一の平面に属するならば直線となることを利用し、同一平面に属すると推定される距離データ点からなる線分を生成する。そして、得られた複数の線分からなる線分群において、同一平面を構成するとされる線分群に基づき平面を推定、検出する方法である。

図43は、線分拡張法による平面検出処理を示すフローチャートである。図43に示

すように、先ず、距離画像を入力し(ステップS71)、距離画像の行方向(又は列方向)の各画素列において同一平面に属すると推定されるデータ点から線分を求める(ステップS72)。そして、これらの線分群の中から同一平面に属すると推定される線分を抽出し、これらの線分からなる平面を求める(ステップS73)。このステップS73では、まず、平面の種となる領域(以下、領域種(seed region)という。)を選び、該当する領域種を選択する。この選択においては、上下隣接する行方向(又は左右隣接する列方向)の1ラインを含む3本の線分が同一平面にあることを条件とする。ここで、選択した3本の線分からなる領域種が属する平面を基準平面とし、3本の線分から平均して求まる平面を求めておく。また、3本の線分からなる領域を基準平面領域とする。

そして、選択した領域種に隣接する行方向(又は列方向)の画素列からなる直線と上記基準平面とが同じ平面であるかどうかを空間的な距離を比較することで判断し、同じ平面である場合には、その隣接する線分を基準平面領域に追加し(領域拡張処理)、追加した線分を含めたものとして上記基準平面を更新し(平面更新処理)、これを平面領域に隣接するデータ列に同一平面の線分が存在しなくなるまで繰り返し行う。そして、以上領域種を検索して平面更新及び領域拡張処理を、種となる領域(3本の線分)が存在しなくなるまで繰り返し実行する。最後に、得られた複数の領域群の中から同一平面を構成するものを連結する。そして、本変形例においては、得られた平面に属する線分群のうち、平面から所定の閾値以上外れる線分を除いて再度平面を求める平面再算出処理をステップS74として更に設け、最終的な平面とするが、詳細は後述する。

ここで、3次元距離データから線分を検出し、これを同一平面毎にまとめた領域を1つの平面とする処理は従来の線分拡張法による平面検出処理であるが、本変形例においては、ステップS72における線分抽出方法が従来とは異なる。すなわち、上述したように、距離データ点から線分を求めて距離データ点にできるだけフィットするように線分を生成しようとしても、距離データの精度に応じて閾値を変更しなければ、over-segmentation又はunder-segmentationなどもの問題が生じてしまう。そこで、本変形例においては、この線分抽出において、距離データの分布を解析することで、距離データの精度、ノイズに応じて適応的に閾値を変更する手法を導入するものと

する。

以下、図43に示す線分拡張法による平面検出方法について更に詳細に説明する。線分抽出部(Line Extraction)2aは、上述したように、ステレオビジョンシステム1からの3次元距離画像を入力とし、距離画像の各列または各行毎に3次元空間内で同一平面上にあると推定される線分を検出する。この線分抽出において、計測ノイズなどによる、上述したover-segmentationやunder-segmentationの問題、すなわち、本来は複数の平面であるのに1つの平面として認識してしまったり、本来は1つの平面であるのに、複数の平面として認識してしまったりする問題を回避するため、データ点の分布に応じて適応的に線分フィッティングさせるアルゴリズム(Adaptive Line Fitting)を導入する。Adaptive Line Fittingは、線分抽出部2aにおいて、先ず比較的大きい閾値を使用して大まかに第1の線分としての線分を抽出し、次に抽出された第1の線分に属するデータ点群から後述する最小二乗法によって得られる第2の線分としての線分に対する該データ点群の分布を解析する。すなわち、同一平面上に存在するか否かを大まかに推定してデータ点群を抽出し、抽出したデータ点群におけるデータ点の分布の偏りがあるか否かを解析して同一平面上に存在しているか否かを再度推定する。

本変形例においては、このデータ点の分布を解析し、データ点群が後述するジグザグ形(zig-zag-shape)に当てはまる場合には、分布に偏りがあるとしてデータ点群を分割する処理を行い、これを繰り返すことによって、データ点群に含まれるノイズに対して適応的に線分の抽出を行うアルゴリズムを使用するものとする。

図44は、線分抽出部2aにおける処理、すなわち、図43におけるステップS72の処理の詳細を示すフローチャートである。まず、線分抽出部2aには、距離データが入力される。入力された距離データのうち、例えば行方向の画素列(データ点列)において、3次元空間上で同一平面上に存在すると推定されるデータ点群を抽出する。3次元空間上で同一平面上に存在すると推定されるデータ点群は、例えば隣接するデータ点の距離が、例えば6cm以下など、データ点間の3次元空間における距離が所定の閾値以下のものからなるデータ点の集合などとして行うことができ、これをデータ点群( $P[0 \cdots n-1]$ )として抽出する(ステップS81)。そして、このデータ点群 $P[0 \cdots$

$n-1]$ に含まれるサンプル数 $n$ が処理に最低限必要なサンプル数(必要最小値) $\min\_n$ より多いか否かをチェックし(ステップS82)、データ数 $n$ が必要最小値 $\min\_n$ より少ない場合(S82:YES)には、検出結果として空集合を出力して処理を終了する。

一方、サンプル数 $n$ が必要最小値 $\min\_n$ 以上である場合(S82:NO)、データ点群 $P[0 \cdots n-1]$ の一方の端点 $P[0]$ と他方の端点 $P[n-1]$ とを結ぶ線分(弦) $L1$ を第1の線分として生成する。そして、データ点群 $P[0 \cdots n-1]$ から、この線分 $L1$ との距離が最も大きいデータ点を着目点 $\text{brk}$ として検索し、その距離 $\text{dist}$ を算出する(ステップS83)。最大距離 $\text{dist}$ がデータ点群分割の閾値 $\max\_d$ より大きい場合には(S84:YES)、データ点群 $P[0 \cdots n-1]$ を着目点(分割点) $\text{brk}$ にて2つのデータ点群 $p[0 \cdots \text{brk}]$ 及び $P[\text{brk} \cdots n-1]$ に分割する(ステップS88)。

一方、最大距離 $\text{dist}$ がデータ点群分割の閾値 $\max\_d$ より小さい場合には(S84:NO)、データ点群 $P[0 \cdots n-1]$ から後述する最小二乗法によって最適な線分の方程式 $\text{line}$ を求め(ステップS85)、この方程式 $\text{line}$ が示す線分 $L2$ を第2の線分として生成する。そして、データ点群 $P[0 \cdots n-1]$ がこの線分 $L2$ に対して後述するZig-Zag-Shapeであるかどうかを調べる(ステップS86)、Zig-Zag-Shapeでない場合(S86:NO)、得られた線分の方程式 $\text{line}$ を線分抽出結果リストに追加し(ステップS87)、処理を終了する。

また、ステップS86においてステップS85で求めた線分がZig-Zag-Shapeである判断された場合(S86:YES)、上述のステップS84と同様、ステップS88に進み、ステップS83において距離 $\text{dist}$ を求めた着目点 $\text{brk}$ にてデータ点群を2つのデータ点群 $P[0 \cdots \text{brk}]$ 及び $P[\text{brk} \cdots n-1]$ に分割する。このステップS88にて2つのデータ点群が得られた場合には、それぞれを再帰的に再度ステップS81からの処理を行う。そして、この処理を分割された全てのデータ点について分割されなくなるまで、すなわち全てのデータ点群がステップS87を経るまで処理を繰り返す、これにより、全ての線分が登録された線分抽出結果リストを得る。このような処理によって、データ点群 $P[0 \cdots n-1]$ からノイズの影響を排除し複数の線分からなる線分群を精度よく検出することができる。

なお、ステップS83にてデータ点群 $P[0 \cdots n-1]$ の端点を結ぶ線分 $L1$ を生成するも

のとして説明したが、例えばデータ点群 $P[0 \cdots n-1]$ の分布、性質など必要に応じてデータ点群 $P[0 \cdots n-1]$ から最小二乗により線分 $L1$ を求めてもよい。また、本変形例においては、着目点 $brk$ は、端点を結んだ線分 $L1$ との距離が最大の点1つとしているが、例えば、上記のように最小二乗により求めた線分との距離が最大の点としたり、距離がデータ点群分割の閾値 $max\_d$ 以上のものが複数ある場合はそれら全ての点又は選択した1つ以上にてデータ点群 $P[0 \cdots n-1]$ を分割するようにしてもよい。

次に、ステップS85における最小二乗による線分生成方法 (Least-Squares Line Fitting) について説明する。ある $n$ 個のデータ点群 $P[0 \cdots n-1]$ が与とき、データ点群に最もフィットした直線の方程式を求める方法を示す。直線の方程式のモデルを下記式(1)で表す。

[数1]

$$x \cos \alpha + y \sin \alpha + d = 0 \quad \cdots (1)$$

この場合、 $n$ 個のデータ点群 $P[0 \cdots n-1]$ の1点 $(x_i, y_i)$ において、直線方程式のモデルとデータ点との誤差の総和は下記式(2)で表すことができる。

[数2]

$$E_{fit} = \sum_i (x_i \cos \alpha + y_i \sin \alpha + d)^2 \quad \cdots (2)$$

データ点群に最もフィットした直線は、上記式(2)の誤差の総和を最小化することによって求められる。上記式(2)を最小にする $\alpha$ 及び $d$ は、データ点群 $P$ の平均及び分散共分散行列を用いて下記(3)のように求めることができる。

[数3]

$$\alpha = \frac{1}{2} \tan^{-1} \frac{-2S_{xy}}{S_{y^2} - S_{x^2}}, \quad d = -(\bar{x} \cos \alpha + \bar{y} \sin \alpha) \quad \cdots (3)$$

ただし、

$$\begin{aligned} \bar{x} &= \frac{1}{n} \sum_i x_i & \bar{y} &= \frac{1}{n} \sum_i y_i \\ S_{x^2} &= \sum_i (x_i - \bar{x})^2 & S_{y^2} &= \sum_i (y_i - \bar{y})^2 \\ S_{xy} &= \sum_i (x_i - \bar{x})(y_i - \bar{y}) \end{aligned}$$



次に、ステップS86におけるジグザグ形(Zig-Zag-Shape)判別方法について説明する。このZig-Zag-Shape判別では、ある $n$ 個のデータ点群 $P[0 \cdots n-1]$ と直線 $Line(\alpha, d)$ 、 $x \cos \alpha + y \cos \alpha + d = 0$ が与えられたとき、そのデータ点群 $P[0 \cdots n-1]$ が、図45Aに示すように直線 $Line$ に対して交差するか、図45Bに示すように、例えばノイズなどの影響によりデータ点が一様に分布しているかを判別するものである。基本的には、直線 $Line$ の一方にデータ点群 $P[0 \cdots n-1]$ が連続して現れる数をカウントし、ある一定数を超えて連続して現れる場合には、zig-zag-shapeであると判断することができる。図45Aの場合には、データ点群 $P[0 \cdots n-1]$ によりよくフィットする直線 $Line$ を求めるためにデータ点群 $P[i]$ を分割する必要がある。図46は、Zig-Zag-Shape判別方法を示すフローチャートである。

まず、データ点群 $P[0 \cdots n-1]$ と直線 $Line(\alpha, d, \sigma)$ とを入力する(ステップS90)。ここで、 $\sigma$ は、点列の標準偏差を示す。次に、この標準偏差 $\sigma$ が所定の閾値 $th\_ \sigma$ より大きいかな否かを判断する。この標準偏差 $\sigma$ が閾値 $th\_ \sigma$ より小さい場合(ステップS91:No)は、演算器の浮動小数点演算誤差による誤差検出の影響を回避するため、判別を終了する。そして、標準偏差 $\sigma$ が閾値 $th\_ \sigma$ より大きい場合のみ判別処理を継続する。次に、データ点群 $P[0 \cdots n-1]$ のうちの最初のデータ点 $P[0]$ が直線のどちら側にあるかを $\text{sing}(\text{sdist}(P[0]))$ によって判断し、この結果を $val_0$ に代入すると共に $val_0$ と同じ側にあるデータ点の連続数をカウントするカウンタ(以下、連続点カウンタといい、このカウント値をカウント値countという。)のカウント値countを1に設定する(ステップS92)。ここで、 $\text{sign}(x)$ は、 $x$ の値の符号(+又は-)を返す関数であり、 $\text{sdist}(i)$ は、 $P[i] \cdot x \cos \alpha + P[i] \cdot y \cos \alpha + d$ として計算された直線 $Line$ において、 $i$ 番目のデータ点との正負の距離を示す。すなわち、 $Val_0$ には、データ点 $P[0]$ が直線 $Line$ のどちら側にあるかで+又は-の符号が代入される。

次に、データ点をカウントするためのカウンタ(以下、データ点カウンタといい、このカウント値をカウント値 $i$ という。)のカウント値 $i$ を1とする(ステップS93)。そして、データ点カウンタのカウント値 $i$ がデータ数 $n$ より小さい場合(ステップS94:YES)、その次のデータ(以下、 $i$ 番目とする。)のデータ点であるデータ点 $P[i]$ が直線のどちら側にあるかを $\text{sing}(\text{sdist}(P[i]))$ によって判断し、この結果を $val$ に代入する(ステップS95)。



そして、ステップ92にて求めた $val_0$ とステップS95にて求めた $val$ とを比較し、 $val_0$ と $val$ とが異なる場合(ステップS96:NO)、 $val_0$ に $val$ を代入し、連続点カウンタのカウント値countに1を代入し(ステップS98)、データ点カウンタのカウント値 $i$ をインクリメントして(ステップS100)ステップS94からの処理に戻る。

一方、ステップS96において、 $val_0$ と $val$ とが同じ場合(ステップS96:YES)、データ点 $P[i-1]$ と $P[i]$ は、直線Lineに対して同じ側にあると判断され、連続点カウンタのカウント値countを1つインクリメントする(ステップS97)。更に、連続点カウンタのカウント値countがZig-Zag-Shapeと判定されるための最小のデータ点数 $min\_c$ より大きいかな否か判定し(ステップS99)、大きい場合には(ステップS99:YES)、Zig-Zag-Shapeと判断し、TRUEを出力して処理を終了する。一方、連続点カウンタのカウント値countが最小のデータ点数 $min\_c$ より小さい場合には(ステップS99:NO)、ステップS100に進み、データ点カウンタのカウント値 $i$ をインクリメントして(ステップS100)、ステップS94からの処理を繰り返す。

そして、このステップS94からの処理を、データ点カウンタのカウント値 $i$ がデータ点数 $n$ に到達するまで続け、カウント値 $i \geq n$ となったところで、FALSEを出力して処理を終了する。

このようなジグザク形判別処理によって、 $n$ 個のデータ点群 $P[0 \cdots n-1]$ と直線Line( $\alpha, d$ ):  $x \cos \alpha + y \sin \alpha + d = 0$ が与えられたとき、このデータ点群が直線Lineに対してzig-zagに交差しているかどうかを判断することができる。これによって、上述したように、ステップS86にてデータ点群を分割するべきかどうかを判断することができ、最小二乗により求めた直線に対し、データ点群がzig-zagに交差していると判断した場合にはデータ点群を分割すべきと判断してステップS88の処理へ進み、着目点brkを分割点としてデータ点群を分割することができる。なお、上記ステップS91～ステップS100までの処理は図47のように表現することも可能である。

また、このようなZig-Zag-Shape判別処理は、演算器のみならずハードウェアで行うことも可能である。図48は、Zig-Zag-Shape判別処理を行う処理部を示すブロック図である。図48に示すように、Zig-Zag-Shape判別処理部20は、 $n$ 個のデータ点群 $P[0 \cdots n-1]$ が入力され、順次各データ点 $P[i]$ が直線Lineの何れ側に位置するかを判別

し、その判別結果Valを出力する方向判別部21と、1つ後のデータと方向判別部21の結果を比較させるための遅延部22と、データ点P[i]における方向判別結果Valとデータ点P[i-1]における方向判別結果Val<sub>0</sub>とを比較する比較部23と、比較部23においてVal=Val<sub>0</sub>の場合に、カウント値をインクリメントする連続点カウンタ24と、連続点カウンタ24のカウント値countと最小データ点数格納部26から読み出した最小データ点数min\_cとを比較する比較部25とを有する。

このZig-Zag-Shape判別処理部における動作は以下になる。すなわち、方向判別部21は、データ点群P[0...n-1]から最小二乗法により直線Lineを求め、各データ点P[i]と直線Lineとの正負の距離を求め、その正負の符号を出力する。遅延部22は、データ点P[i-1]の直線Lineまでの距離に対する正負の符号が入力されると1つ後のデータ点P[i]の正負の符号が入力されるタイミングまでデータを格納する。

比較部23は、データ点P[i]とデータ点P[i-1]の上記正負の符号を比較し、同じ符号である場合にはカウンタ24のカウント値countをインクリメントする信号を出力し、正負の符号が異なればカウント値countに1を代入する信号を出力する。比較部25は、カウント値countと最小データ点数min\_cとを比較し、最小データ点数min\_cよりカウント値countが大きい場合には、データ点群P[0...n-1]がジグザグであることを示す信号を出力する。

次に、図40に示す領域拡張部 (Region Growing) 2bについて説明する。領域拡張部2bは、線分抽出部2aによって得られた線分群を入力とし、それらの線分それぞれがどの平面に属しているかを点列の平面への当てはめ (Plane Fitting) により判断し、与えられる線分群からなる領域を複数の平面 (平面領域) に分離する。複数の平面に分離するために、以下の手法をとる。

まず、与えられた線分群から、同じ平面上にあると推定される隣接する3本の線分を検索する。この3本の線分により求められる平面 (基準平面) が、平面の種となるものであり、この3本の線分が含まれる領域を領域種 (seed region) という。そして、この領域種に隣接する線分を順次、基準平面と同一平面上にある線分か否かを点列の平面への当てはめ (Plane Fitting) により判断し、隣接する線分が同じ平面に含まれると判断された場合には、この線分を領域拡大用の線分として領域種に追加してその

領域を拡大すると共に、基準平面の方程式を上記領域拡大用の線分を含めて再度算出し直す。このような処理によって、全ての線分を何れかの領域(平面)に配分する。

図49は、領域拡張処理を説明するための模式図である。図49に示すように、画像30内に複数の平面からなる階段31が存在する場合、例えば太線で示す32a〜32cの3本の線分が領域種として選択されたとする。これら3本の線分32a〜32cからなる領域が領域種となる。まず、この3つの線分32a〜32cにより1つの平面(基準平面)Pを求める。次に、領域種の最も外側の線分32a又は32cに領域種外にて隣接するそれぞれデータ列33又は34において、平面Pと同一の平面である線分を選択する。ここでは、線分33aが選択されたとする。次に、これら4本の線分群からなる平面P'を求め、基準平面Pを更新する。次に、線分34aが選択されれば、5本の線分群からなる平面P''を求め、平面P'を更新する。これを繰り返すことにより、階段31の2段目の踏面が、破線で囲まれる平面45として求められる。このようにして、選択された領域種を種として追加する線分がなくなるまで領域拡大処理する。そして、追加する線分がなくなった場合、再び画像30内から領域種となる3つの線分を検索して領域拡大処理を実行するというような処理を繰り返し、領域種となる3つの線分がなくなるまで図43のステップS3の処理を繰り返す。

次に、データ点群 $P[0 \cdots n-1]$ から構成される平面の方程式を推定する手法(Plane Fitting)、これを使用して領域種を選択する方法(Selection of seed region)、領域種から領域を拡大していく領域拡張処理(Region growing)、及び得られた平面方程式から誤差が大きいものなどを除いて再度算出する後処理(Post processing)について説明する。

3次元空間内の点Pは $P = (x_i, y_i, z_i)$ により表され、平面の方程式はその法線ベクトル $n(n_x, n_y, n_z)$ と非負の定数 $d$ によって下記式(4)で表される。

[数4]

$$xn_x + yn_y + zn_z + d = 0 \quad \cdots (4)$$

ここで、ステレオカメラでは、焦点を通る平面を観測することができない、すなわち、

平面は焦点を通らないため、 $d \neq 0$ とすることができる。したがって、平面は、最小二乗法により下記式(5)に示す値を最小にする値として求めることができる。

[数5]

$$fit(n, d) = \sum_i (p_i^T n + d)^2 \quad \dots(5)$$

最適解は $n = m / \|m\|$ ,  $d = -1 / \|m\|$ として求まる。ここで、 $\|m\|$ は、ベクトルの大きさ、 $m$ は、行列式によって連立一次方程式を解くクラメールの法則(Cramer's rule)を使用して下記(6-1)のように容易に得られる線形システムの解である。

[数6]

$$A \cdot m = b \quad \dots(6-1)$$

ここで、

$$A = \sum_i p_i p_i^T, \quad b = \sum_i p_i \quad \dots(6-2)$$

この解は、新たなデータ点が加えられたり、又はデータ点削除されたりした場合であっても、上記式(6-2)に示す $A$ と $b$ の値を更新するのみで、平面パラメータを再計算することができる。更に、本変形例における線分抽出方法の場合は $n$ 個のデータ点群の2つのモーメント(1次モーメント:平均、2次モーメント:分散) $E(p)$ 、 $E(pp_T)$ が既知であり、これらを使用して、下記(7)に示すように $A$ 、 $b$ を更新することができ、 $n$ 個のデータ点群における平面更新処理に拡張することができる。

[数7]

$$A \leftarrow A + nE(pp^T), \quad b \leftarrow b + nE(p) \quad \dots(7)$$

また、一度平面パラメータ $n$ 、 $d$ を算出すれば、求まった平面方程式から、 $n$ 個のデータ点群の平面方程式からの外れ度合いを示す平面方程式の2乗平均誤差(RMS (root mean square) residual) (以下、rmsという。)を下記式(8)により算出することができる。この場合も、 $n$ 個のデータ点の上記2つのモーメントを使用して下記式(8)を求めることができる。

[数8]

$$rms(p_1 \cdots p_n) = \sqrt{\frac{1}{n} \sum_i (p_i^T n + d)^2} > th\_rms \quad \cdots (8)$$

上記(8)に示すように、各データ点が求めた平面上にあれば平面方程式の2乗平均誤差 $rms(p_1 \cdots p_n)$ は0になる値であり、この値が小さいほど各データ点が平面によくフィットしていることを示す。

次に、領域種(seed region)を検索する方法及び領域種から領域を拡大すると共に平面を更新する方法について説明する。図50は、領域種を検索する処理及び領域拡張処理の手順を示すフローチャートである。図50に示すように、領域種を選択には、まず、線分抽出の際に使用した行方向又は列方向のデータ列が隣接する3つの線分 $(l_1, l_2, l_3)$ であって、互いの線分 $(l_1, l_2)$ ,  $(l_2, l_3)$ における画素位置が上記データ列とは直交する方向にて重複したものを検索する(ステップS101)。各データ点は画像内における画素位置を示すインデックス(index)を有しており、例えば行方向のデータ列における線分である場合、このインデックスを比較して列方向にて重複しているか否かを比較する。この検索に成功した場合(ステップS102: YES)、上記式(7)を使用して上記(6-1)を算出する。これにより、平面パラメータ $n, d$ を決定でき、これを使用して上記式(8)に示す平面方程式の2乗平均誤差 $(l_1, l_2, l_3)$ を計算する(ステップS103)。そして、この平面方程式の2乗平均誤差 $rms(l_1, l_2, l_3)$ が例えば1cmなどの所定の閾値 $th\_rms$ より小さい場合には、この3つの線分を領域種として選択する(ステップS104)。所定の閾値 $th\_rms$ より大きい場合には、再びステップS101に戻り、上記条件を満たす線分を検索する。また、領域種に選ばれた線分は、線分群のリストから除くことで、他の平面拡張などの際に使用されないようにしておく。

こうして選択された領域種から線分拡張法により領域を拡張する。すなわち、まず、領域種の領域に追加する候補となる線分を検索する(ステップS105)。なお、この領域は、領域種が既に更新されている場合の、後述する更新された領域種も含む。候補となる線分は、領域種の領域に含まれる線分(例えば $l_1$ )に隣接する線分( $l_4$ )であって、上述同様、これらの線分の画素位置が相互に重なりあうことを条件とする。検索が成功した場合(ステップS106: YES)、その平面方程式の2乗平均誤差 $rms(l_4)$ を



算出し、これが所定の閾値 $th_{rms}$ より小さいか否かを判定し(ステップS107)、小さい場合には平面パラメータを更新し(ステップS108)、再びステップS105からの処理を繰り返す。ここで、候補となる線分がなくなるまで処理を繰り返し、候補となる線分がなくなったら(ステップS106:NO)、ステップS101の処理に戻り、再び領域種を検索する。そして、線分群に含まれる領域種がなくなった場合(ステップS102:NO)、今まで得られている平面パラメータを出力して処理を終了する。

ここで、本変形例においては、領域種を検索し、3つの線分が同一平面に属するか否かの判定、及び領域拡張処理を行う際に基準平面又はこれを更新した更新平面に属するか否かの判定には、上記式(8)を使用する。すなわち、平面方程式の2乗平均誤差rmsが所定の閾値( $th_{rms}$ )未満である場合にのみその線分(群)を同一平面に属するものと推定し、その線分を含めた平面として再び平面を算出する。このように平面方程式の2乗平均誤差rmsを使用して同一平面に属するか否かを判定することにより、更にノイズにロバストでかつ、細かい段差を含んでいるような場合にも正確に平面を抽出することができる。以下にその理由について説明する。

図51は、その効果を示す図であって、端点と直線との距離が等しくても平面方程式の2乗平均誤差rmsが異なる例を示す模式図である。ここで、非特許文献4のように、領域拡張処理する際、注目の直線(線分)の端点(end point)と平面Pとの距離Dの値が所定の閾値より小さい場合に、当該注目の線分が平面Pと同一平面であるとして領域拡張処理を行うと、平面Pに交差する直線La(図51A)と、平面Pと平行で所定距離ずれているような直線Lb(図51B)とが同様に平面Pの更新に使用されることとなる。ここで、平面方程式の2乗平均誤差rmsを求めると、図51Aの直線Laから求まる平面方程式の2乗平均誤差rms(La)に比して図51Bの直線Lbから求まる平面方程式の2乗平均誤差rms(Lb)の方が大きい。すなわち、図51Aのように、直線Laと平面Pとが交差する場合は、平面方程式の2乗平均誤差rmsが比較的小さくノイズの影響である場合が多いのに対し、図51Bのような場合、平面方程式の2乗平均誤差rmsが大きく、直線Lbは平面Pと同一平面ではなく異なる平面P'である確率が高い。したがって、複数の平面が含まれるような環境から平面を精確に求める必要がある場合などにおいては、本変形例のように、平面方程式の2乗平均誤差rmsを算出し、この



値が所定の閾値未満である場合に同一平面と判断することが好ましい。なお、環境や距離データの性質に応じて、従来と同様、線分の端点と平面との距離が所定の閾値以下の場合は当該線分を平面に含めるようにしたり、これらを組み合わせてもよい。

また、面パラメータ $n, d$ を一旦算出すれば、平面方程式の2乗平均誤差 $rms$ は、データ点群について線分抽出の際に求めた2つのモーメントの値から平面方程式を更新し、上記式(8)にて簡単に算出することができる。

また、上述の領域種の選択方法は、図52のようにも表現することができる。 $overlap(l_j, l_k)$ は、各イメージロウに含まれる直線ベクトル $l_j$ と $l_k$ における端点間の位置が直線ベクトルとは直交する位置にて重なっている場合に $true$ を出力する関数である。また、 $fitPlane(l_1, l_2, l_3)$ は、上記式(4)～(7)により $Am=b$ の解を求め平面パラメータ $n, d$ を計算し、上記式(8)により算出された $A, b$ により、直線ベクトル $l_1, l_2, l_3$ を平面にフィッティングさせる関数である。

$rms(l_1, l_2, l_3)$ は、上記式(6)を使用して3本の直線全てにおいて、平面方程式の2乗平均誤差 $rms$ の値を算出する関数である。また、 $remove(l_1, l_2, l_3)$ は、 $lines[i], lines[i+1]l_2, lines[i+2]$ から領域種を構成するとして選択されたそれぞれ直線 $l_1, l_2, l_3$ を除くことを意味し、これにより、再びこれらの直線が計算に使用されることを防止する。

また、領域拡張処理は、図53のように表現することもできる。図53において、 $A$ 及び $b$ は、上記式(6-1)に示すそれぞれ行列及びベクトルである、また、 $add(A, b, l)$ は、上記式(8)により、 $A$ と $b$ に直線 $line$ のモーメントを加える関数である。 $Solve(A, b)$ は、 $Am=b$ を満たす $m$ を求め、上記式(4)～(7)により平面パラメータ $n, d$ を計算する。 $select(open)$ は、例えば最初の1つなど、 $open$ の中から任意に1つの要素を選択する関数である。また、 $index(l_i)$ は、画素列又は行における $l_i$ のインデックスを返す関数である。また、 $neighbor(index)$ は、与えられたインデックスに隣接したインデックス、例えば $\{index-1, index+1\}$ を返す関数である。

また、上述したように、本変形例においては、図43のステップS73において領域拡張処理を行って平面方程式を更新した後、ステップS74において平面方程式を再度算出する処理(Post processing)を行う。この再度算出する処理では、例えば上述の

ように更新され最終的に得られた平面方程式が示す平面に属するとされた距離データ点又は線分の平面からのずれを計算し、所定の値以上平面から外れる距離データ点又は線分は除き、再度平面方程式を更新することで、ノイズの影響を更に低減することができる。

次に、このステップS74について詳細に説明する。ここでは、2つのステップにより、平面方程式を再度算出する方法について説明する。まず、ステップS73にて検出された各平面の境界の距離データ点(pixels)において、現在属している平面よりも、隣接する平面までの距離が近いデータ点が検出された場合は、当該データ点を隣接する平面の方に含める処理をする。また、何れの平面にも属していなく、かつ距離が例えば1.5cmなど比較的大きい閾値以下である平面が存在するデータ点が検出できた場合は、当該データ点をその平面に含める処理をする。これらの処理は各平面領域の境界近傍のデータ点を検索することで実行することができる。以上の処理が終了したら、再度平面方程式を算出する。

次に、上述のようにして再度算出された平面の各領域の境界近傍において、各データ点と平面との距離が例えば0.75cmなど比較的小さい閾値を超える場合は、それらのデータ点を捨てる処理を実行する。これにより、その平面領域は若干小さくなるものの更に精確な平面を求めることができる。距離データ点を削除後、再び平面を求め、この処理を繰り返す。このことにより、極めて精密に平面を求めることができる。

次に各処理によって得られる結果を示す。図54Aは、ロボット装置が立った状態で床面を見下ろした際の床面を示す模式図、図54Bは、縦軸をx、横軸をy、各データ点の濃淡でz軸を表現して3次元距離データを示す図であり、更に、行方向の画素列から線分抽出処理にて同一平面に存在するとされるデータ点群から直線を検出したものを示す。図54Bに示す直線群から領域拡張処理によりえられた平面領域を図54Cに示す。このように、ロボット装置の視野内には、1つの平面(床面)のみが存在する、すなわち、床面が全て同じ平面として検出されていることがわかる。

次に、床面に段差を一段置いたときの結果を図55に示す。図55Aに示すように、床面Fには、1段の段差ST3が載置されている。図55Bは、実験条件を示す図であり、着目点と直線(線分)との距離が $\max\_d$ を超える場合は、データ点群を分割する。ま

た、抽出の成否(水平) (correct extraction(horizontal))は、行方向のデータ列毎に、合計10回の線分抽出を行う線分拡張法による平面検出を行って成功した回数を示し、抽出の成否(垂直) (correct extraction(vertical))は、列方向のデータ列毎についての抽出の成否を示す。また、No. 1ーNo. 5は、上述したZig-Zag-Shape判別処理を取り入れていない従来の線分拡張法による平面検出処理の条件、No. 6は、Zig-Zag-Shape判別処理を行った本変形例における平面検出方法の条件を示す。

図55C及び図55Dは、線分拡張法により平面検出した結果を示す図であって、それぞれ本変形例における手法により平面検出した結果、従来の線分拡張法により平面検出した結果(比較例)を示す。図55Bに示すように、従来の手法においては、線分抽出(Line Fitting)において推定のための閾値パラメータmax\_dを大きくする(max\_d=25, 30)と検出率が下がり、閾値max\_d小さくする(max\_d=10, 15)と検出率が向上する。これに対して、本発明のように、ジグザグ形検証処理を導入することにより、大きな閾値max\_d=30を設定しても、優れた検出結果を示すことがわかる。

すなわち、閾値max\_dを大きくすると、ノイズの影響が少なくなるものの、線分抽出が難しくなり、閾値max\_dを小さくすると、ノイズの影響を受けて誤検出が多くなってしまう。図56Bに示す床面を撮影した画像から3次元距離データを取得した場合を図56B及び図56Cに示す。何れ左図は、行方向の画素列(距離データ列)から線分を抽出した例、右図は列方向の画素列(距離データ列)から線分を抽出した例を示す。図56Bに示すように、閾値max\_dを小さくすると、ノイズの影響が大きくなり、ノイズの影響が大きい遠方などにおいては特に、線分をうまく検出することができない。一方、図56Cに示すように、従来の線分抽出に更にジグザグ形判別処理を加えた場合、閾値max\_dを大きくしても、更にノイズの影響が大きい遠方の領域であっても線分が検出されていることがわかる。

これにより、上述した如く、それぞれ異なる階段を撮影した画像から3次元距離データを取得して平面検出することができる。例えば、図11及び図12に示すように、何れの場合も全ての踏面を平面として検出できている。なお、図12Bでは、床面の一部も他の平面として検出成功している。

本変形例によれば、線分拡張法による平面検出を行う際、始めは大きな閾値を設

定して線分を分割し、次にZig-Zag-Shape判別処理により、閾値を超えるデータ点を持たない直線であってもジグザグ形である場合には、ノイズではなく、複数平面からなる直線であるとして線分を分割するようにしたので、ノイズを含む距離情報から複数の平面を精度よく検出することが可能となる。

このように、小さい段差も精度よく検出することができるため、例えばロボット装置が移動可能な環境内の階段などを認識することができ、二足歩行ロボット装置であれば、この検出結果を利用して階段昇降動作が可能となる。

更に、複数の平面によって構成されている凸凹の床面を歩行可能な平面だと誤認識することがなくなり、ロボット装置の移動などが更に簡単になる。

なお、本発明は上述した変形例のみに限定されるものではなく、本発明の要旨を逸脱しない範囲において種々の変更が可能であることは勿論である。また、上述した平面検出処理、階段認識処理、階段昇降制御処理のうち1以上の任意の処理は、ハードウェアで構成しても、演算器(CPU)にコンピュータプログラムを実行させることで実現してもよい。コンピュータプログラムとする場合には、記録媒体に記録して提供することも可能であり、また、インターネットその他の伝送媒体を介して伝送することにより提供することも可能である。

## 請求の範囲

- [1] 1. 移動手段により移動可能なロボット装置において、  
3次元の距離データから環境内に含まれる1又は複数の平面を検出し、平面情報として出力する平面検出手段と、  
上記平面情報から移動可能な平面を有する階段を認識し、該階段の踏面に関する踏面情報及び蹴り上げ情報を有する階段情報を出力する階段認識手段と、  
上記階段情報に基づき、階段昇降可能か否かを判断し、昇降動作が可能であると判断した場合には、その踏面に対して自律的に位置決めして階段昇降動作を制御する階段昇降制御手段を有する  
ことを特徴とするロボット装置。
- [2] 2. 上記3次元の距離データを取得する距離計測手段を有する  
ことを特徴とする請求の範囲第1項記載のロボット装置。
- [3] 3. 上記移動手段として脚部を備え、この脚部により移動可能である  
ことを特徴とする請求の範囲第1項記載のロボット装置。
- [4] 4. 上記階段認識手段は、  
与えられた平面情報から移動可能な平面を有する階段を検出して統合前階段情報を出力する階段検出手段と、  
上記階段検出手段から出力される時間的に異なる複数の統合前階段情報を統計的に処理することにより統合した統合済階段情報を上記階段情報として出力する階段統合手段とを有する  
ことを特徴とする請求の範囲第1項記載のロボット装置。
- [5] 5. 上記階段検出手段は、上記平面情報に基づき踏面の大きさ及び空間的な位置を認識し、この認識結果である踏面情報を上記統合前階段情報として出力し、  
上記階段統合手段は、時間的に前後する踏面情報から、所定の閾値より大きい重複領域を有しかつ相対的な高さの違いが所定の閾値以下である2以上の踏面からなる踏面群を検出した場合、当該踏面群を何れをも含む一の踏面となるよう統合する  
ことを特徴とする請求の範囲第4項記載のロボット装置。
- [6] 6. 上記階段認識手段は、上記平面情報に基づき踏面の大きさ及び空間的な位置を



認識し上記踏面情報とする

ことを特徴とする請求の範囲第1項記載のロボット装置。

- [7] 7. 上記踏面情報は、少なくとも移動方向に対して該踏面の手前側の境界を示すフロントエッジ及び奥側の境界を示すバックエッジの情報を含む

ことを特徴とする請求の範囲第6項記載のロボット装置。

- [8] 8. 上記踏面情報は、上記フロントエッジ及びバックエッジに挟まれた領域である安全領域の左右両側に隣接した領域であって移動可能である確率が高いと推定されるマージン領域を示す右側マージン情報及び左側マージン情報を有する

ことを特徴とする請求の範囲第7項記載のロボット装置。

- [9] 9. 上記踏面情報は、上記平面情報に基づき踏面と推定された領域の重心を示す参照点情報を有する

ことを特徴とする請求の範囲第7項記載のロボット装置。

- [10] 10. 上記参照点情報は、上記フロントエッジ及びバックエッジに挟まれた領域を示す安全領域の重心、該安全領域及びその両側に隣接した領域であって移動可能である確率が高いと推定されるマージン領域からなる踏面領域の重心、又は踏面となる平面を構成する点群から求まる中心点の何れかからなる参照点の位置情報を有する

ことを特徴とする請求の範囲第9項記載のロボット装置。

- [11] 11. 上記踏面情報は、踏面となる平面を構成する点群の3次元座標情報を有する

ことを特徴とする請求の範囲第7項記載のロボット装置。

- [12] 12. 上記階段認識手段は、上記平面情報に基づき平面の境界を抽出して多角形を算出し、該多角形に基づき上記踏面情報を算出する

ことを特徴とする請求の範囲第1項記載のロボット装置。

- [13] 13. 上記多角形は、上記平面情報に基づき抽出された平面の境界に外接する凸多角形領域である

ことを特徴とする請求の範囲第12項記載のロボット装置。

- [14] 14. 上記多角形は、上記平面情報に基づき抽出された平面の境界に内接する凸多角形領域である

ことを特徴とする請求の範囲第12項記載のロボット装置。



- [15] 15. 上記多角形は、上記平面情報に基づき抽出された平面の境界を平滑化した非凸多角形領域であることを特徴とする請求の範囲第12項記載のロボット装置。
- [16] 16. 上記平面情報は、一の平面毎に、法線ベクトル、境界を示す境界情報、重心位置を示す重心情報、大きさを示す面積情報、及び平面度から選択される1以上の情報を有する  
ことを特徴とする請求の範囲第1項記載のロボット装置。
- [17] 17. 上記階段昇降制御手段は、現在移動中の移動面におけるバックエッジに対峙した所定位置に移動した後、昇降動作を実行するよう制御する  
ことを特徴とする請求の範囲第3項記載のロボット装置。
- [18] 18. 上記階段昇降制御手段は、現在移動中の移動面におけるバックエッジが確認できない場合は、次に昇降動作の対象となる次段の踏面におけるフロントエッジに対峙した所定位置に移動した後、昇降動作を実行するよう制御する  
ことを特徴とする請求の範囲第17項記載のロボット装置。
- [19] 19. 上記階段昇降制御手段は、次に移動対象となる踏面を検出し、当該移動対象となる踏面に対峙した所定位置に移動する一連の動作を行って昇降動作を実行するよう制御する  
ことを特徴とする請求の範囲第3項記載のロボット装置。
- [20] 20. 上記階段昇降制御手段は、現在位置から次に移動対象となる次段又は次段以降の踏面が検出できない場合、過去に取得した階段情報から当該移動対象となる次段の踏面を検索する  
ことを特徴とする請求の範囲第19項記載のロボット装置。
- [21] 21. 上記階段昇降制御手段は、現在の移動面におけるバックエッジに対峙した所定位置に移動した後、次の移動対象となる踏面を検出し、当該踏面におけるフロントエッジに対峙した所定位置に移動し、当該踏面に移動する昇降動作を実行するよう制御する  
ことを特徴とする請求の範囲第3項記載のロボット装置。
- [22] 22. 上記階段昇降制御手段は、踏面に対する上記移動手段の位置を規定したパラメータを使用して昇降動作を制御する

ことを特徴とする請求の範囲第3項記載のロボット装置。

- [23] 23. 上記パラメータは、上記脚部の足上げ高さ又は足下げ高さに基づき決定されることを特徴とする請求の範囲第22項記載のロボット装置。

- [24] 24. 階段を登る動作と降りる動作とで上記パラメータの数値を変更するパラメータ切り替え手段を有する

ことを特徴とする請求の範囲第22項記載のロボット装置。

- [25] 25. 上記平面検出手段は、3次元空間で同一平面上にあると推定される距離データ点群毎に線分を抽出する線分抽出手段と、上記線分抽出手段によって抽出された線分群から同一平面に属すると推定される複数の線分を抽出し該複数の線分から平面を算出する平面領域拡張手段とを有し、

上記線分抽出手段は、距離データ点の分布に応じて適応的に線分を抽出する

ことを特徴とする請求の範囲第1項記載のロボット装置。

- [26] 26. 上記線分抽出手段は、上記距離データ点間の距離に基づき同一平面上にあると推定される距離データ点群を抽出し、該距離データ点群における距離データ点の分布に基づき、当該距離データ点群が同一平面上にあるか否かを再度推定する

ことを特徴とする請求の範囲第25項記載のロボット装置。

- [27] 27. 上記線分抽出手段は、上記同一平面上にあると推定される距離データ点群から線分を抽出し、該距離データ点群のうち該線分との距離が最も大きい距離データ点を着目点とし、当該距離が所定の閾値以下である場合に該距離データ点群における距離データ点の分布に偏りがあるか否かを判別し、該偏りがある場合には該着目点にて該距離データ点群を分割する

ことを特徴とする請求の範囲第25項記載のロボット装置。

- [28] 28. 上記線分抽出手段は、上記同一平面上にあると推定される距離データ点群から第1の線分を抽出し、該距離データ点群のうち該第1の線分との距離が最も大きい距離データ点を着目点とし、当該距離が所定の閾値以下である場合に該距離データ点群から第2の線分を抽出し、該第2の線分の一方側に距離データ点が所定の数以上連続して存在するか否かを判定し、所定の数以上連続して存在する場合に該距離データ点群を該着目点にて分割する

- ことを特徴とする請求の範囲第25項記載のロボット装置。
- [29] 29. 上記平面領域拡張手段は、同一の平面に属すると推定される1以上の線分を選択して基準平面を算出し、該基準平面と同一平面に属すると推定される線分を該線分群から拡張用線分として検索し、該拡張用線分により該基準平面を更新すると共に該基準平面の領域を拡張する処理を繰り返し、更新が終了した平面を更新済平面として出力する
- ことを特徴とする請求の範囲第25項記載のロボット装置。
- [30] 30. 上記更新済平面に属する距離データ点群において、当該更新済平面との距離が所定の閾値を超える距離データ点が存在する場合、これを除いた距離データ点群から再度平面を算出する平面再算出手段を更に有する
- ことを特徴とする請求の範囲第29項記載のロボット装置。
- [31] 31. 上記平面領域拡張手段は、線分により定まる平面と上記基準平面との誤差に基づき当該線分が該基準平面と同一平面に属するか否かを推定する
- ことを特徴とする請求の範囲第25項記載のロボット装置。
- [32] 32. 移動手段により移動可能なロボット装置の動作制御方法において、  
3次元の距離データから環境内に含まれる1又は複数の平面を検出し、平面情報として出力する平面検出工程と、  
上記平面情報から移動可能な平面を有する階段を認識し、該階段の踏面に関する踏面情報及び蹴り上げ情報を有する階段情報を出力する階段認識工程と、  
上記階段情報に基づき、階段昇降可能か否かを判断し、昇降動作が可能であると判断した場合には、その踏面に対して自律的に位置決めして階段昇降動作を制御する階段昇降制御工程と  
を有することを特徴とするロボット装置の動作制御方法。
- [33] 33. 上記3次元の距離データを取得する距離計測工程を有する
- ことを特徴とする請求の範囲第32項記載のロボット装置の動作制御方法。
- [34] 34. 上記移動手段は、機体脚部である
- ことを特徴とする請求の範囲第32項記載のロボット装置の動作制御方法。
- [35] 35. 上記階段認識工程は、

与えられた平面情報から移動可能な平面を有する階段を検出して統合前階段情報を出力する階段検出工程と、

上記階段検出工程にて出力される時間的に異なる複数の統合前階段情報を統計的に処理することにより統合した統合済階段情報を上記階段情報として出力する階段統合工程とを有する

ことを特徴とする請求の範囲第32項記載のロボット装置の動作制御方法。

- [36] 36. 上記階段検出工程では、上記平面情報に基づき踏面の大きさ及び空間的な位置を認識し、この認識結果である踏面情報を上記統合前階段情報として出力し、

上記階段統合工程では、時間的に前後する踏面情報から、所定の閾値より大きい重複領域を有しかつ相対的な高さの違いが所定の閾値以下である2以上の踏面からなる踏面群を検出した場合、当該踏面群を何れも含む一の踏面となるよう統合する

ことを特徴とする請求の範囲第35項記載のロボット装置の動作制御方法。

- [37] 37. 上記階段認識工程では、上記平面情報に基づき踏面の大きさ及び空間的な位置を認識し上記踏面情報とする

ことを特徴とする請求の範囲第32項記載のロボット装置の動作制御方法。

- [38] 38. 上記踏面情報は、少なくとも移動方向に対して該踏面の手前側の境界を示すフロントエッジ及び奥側の境界を示すバックエッジを示す情報を含む

ことを特徴とする請求の範囲第37項記載のロボット装置の動作制御方法。

- [39] 39. 上記階段認識工程では、上記平面情報に基づき平面の境界を抽出して多角形を算出し、該多角形に基づき上記踏面情報を算出する

ことを特徴とする請求の範囲第32項記載のロボット装置の動作制御方法。

- [40] 40. 上記階段昇降制御工程では、現在移動中の移動面におけるバックエッジに対峙した所定位置に移動した後、昇降動作を実行するよう制御する

ことを特徴とする請求の範囲第34項記載のロボット装置の動作制御方法。

- [41] 41. 上記階段昇降制御工程では、現在移動中の移動面におけるバックエッジが確認できない場合は、次に昇降動作の対象となる次段の踏面におけるフロントエッジに対峙した所定位置に移動した後、昇降動作を実行するよう制御する

ことを特徴とする請求の範囲第40項記載のロボット装置の動作制御方法。



- [42] 42. 上記階段昇降制御工程では、次に移動対象となる踏面を検出し、当該踏面に対峙した所定位置に移動する一連の動作を行って昇降動作を実行するよう制御することを特徴とする請求の範囲第34項記載のロボット装置の動作制御方法。
- [43] 43. 上記階段昇降制御工程では、現在位置から次に移動対象となる次段又は次段以降の踏面が検出できない場合、過去に取得した階段情報から当該移動対象となる次段の踏面を検索することを特徴とする請求の範囲第42項記載のロボット装置の動作制御方法。
- [44] 44. 上記階段昇降制御工程では、現在の移動面におけるバックエッジに対峙した所定位置に移動した後、次の移動対象となる踏面を検出し、当該踏面におけるフロントエッジに対峙した所定位置に移動し、当該踏面に移動する昇降動作を実行するよう制御することを特徴とする請求の範囲第34項記載のロボット装置の動作制御方法。
- [45] 45. 上記階段昇降制御工程では、踏面に対する移動手段の位置を規定するためのパラメータを使用して昇降動作を制御することを特徴とする請求の範囲第34項記載のロボット装置の動作制御方法。
- [46] 46. 階段を登る動作と降りる動作とで上記パラメータの数値を変更するパラメータ切り替え工程を有することを特徴とする請求の範囲第45項記載のロボット装置の動作制御方法。
- [47] 47. 移動手段により移動可能な移動装置において、  
3次元の距離データから環境内に含まれる1又は複数の平面を検出し、平面情報として出力する平面検出手段と、  
上記平面情報から移動可能な平面を有する階段を認識し、該階段の踏面に関する踏面情報及び蹴り上げ情報を有する階段情報を出力する階段認識手段と、  
上記階段情報に基づき、階段昇降可能か否かを判断し、昇降動作が可能であると判断した場合には、その踏面に対して自律的に位置決めして階段昇降動作を制御する階段昇降制御手段とを有することを特徴とする移動装置。

[図1]

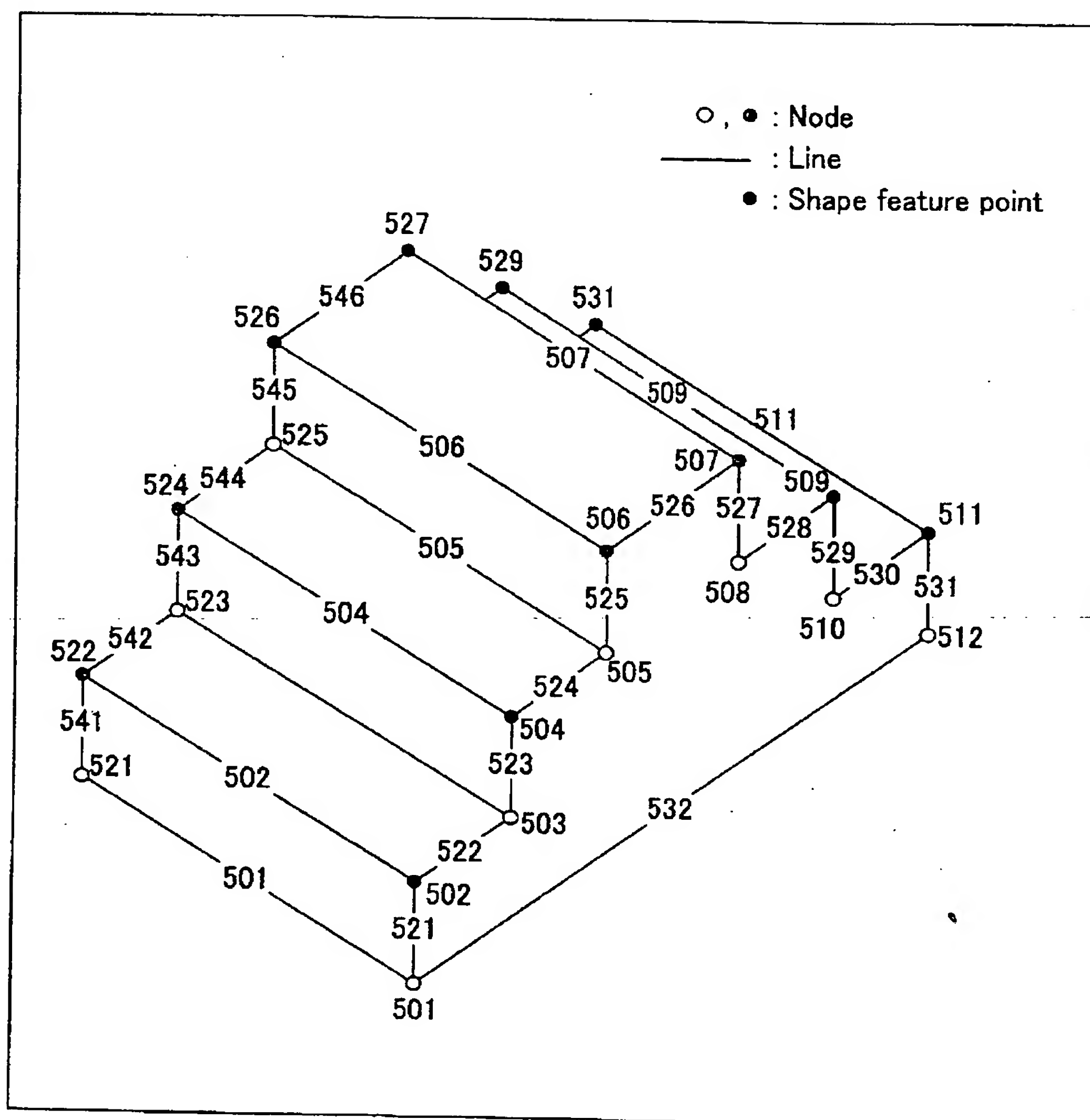


FIG.1



[図2]

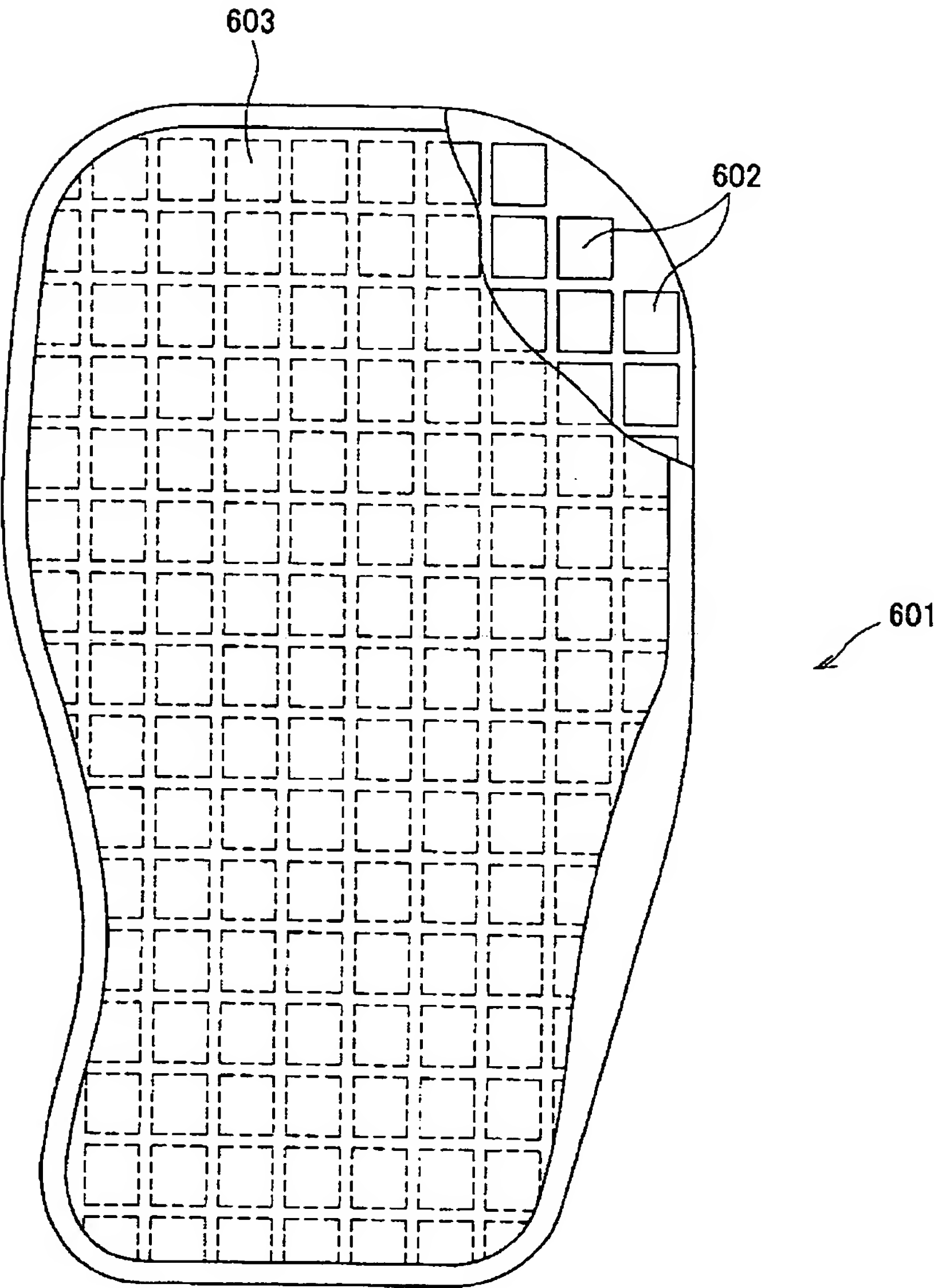


FIG.2

[図3]

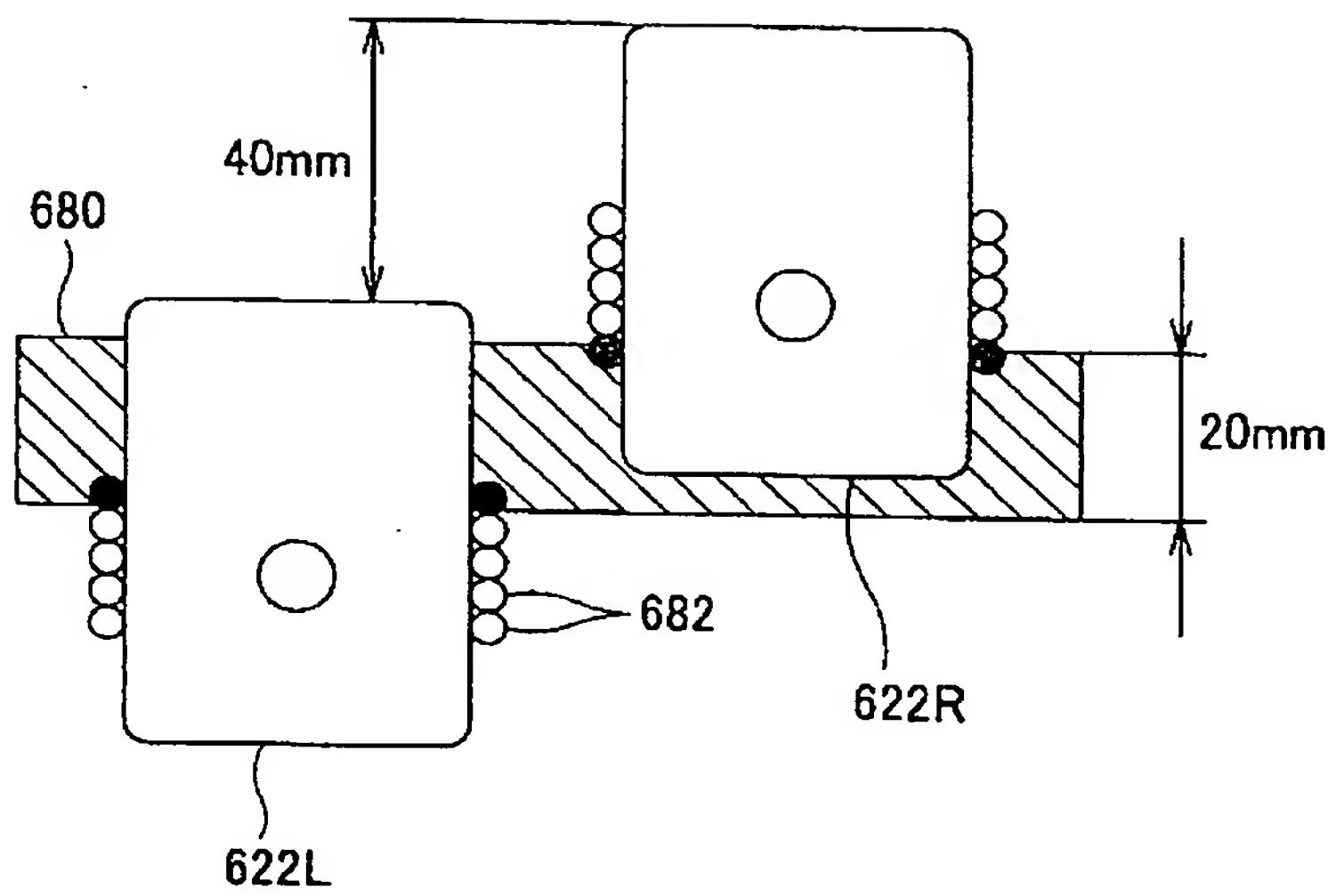


FIG.3A

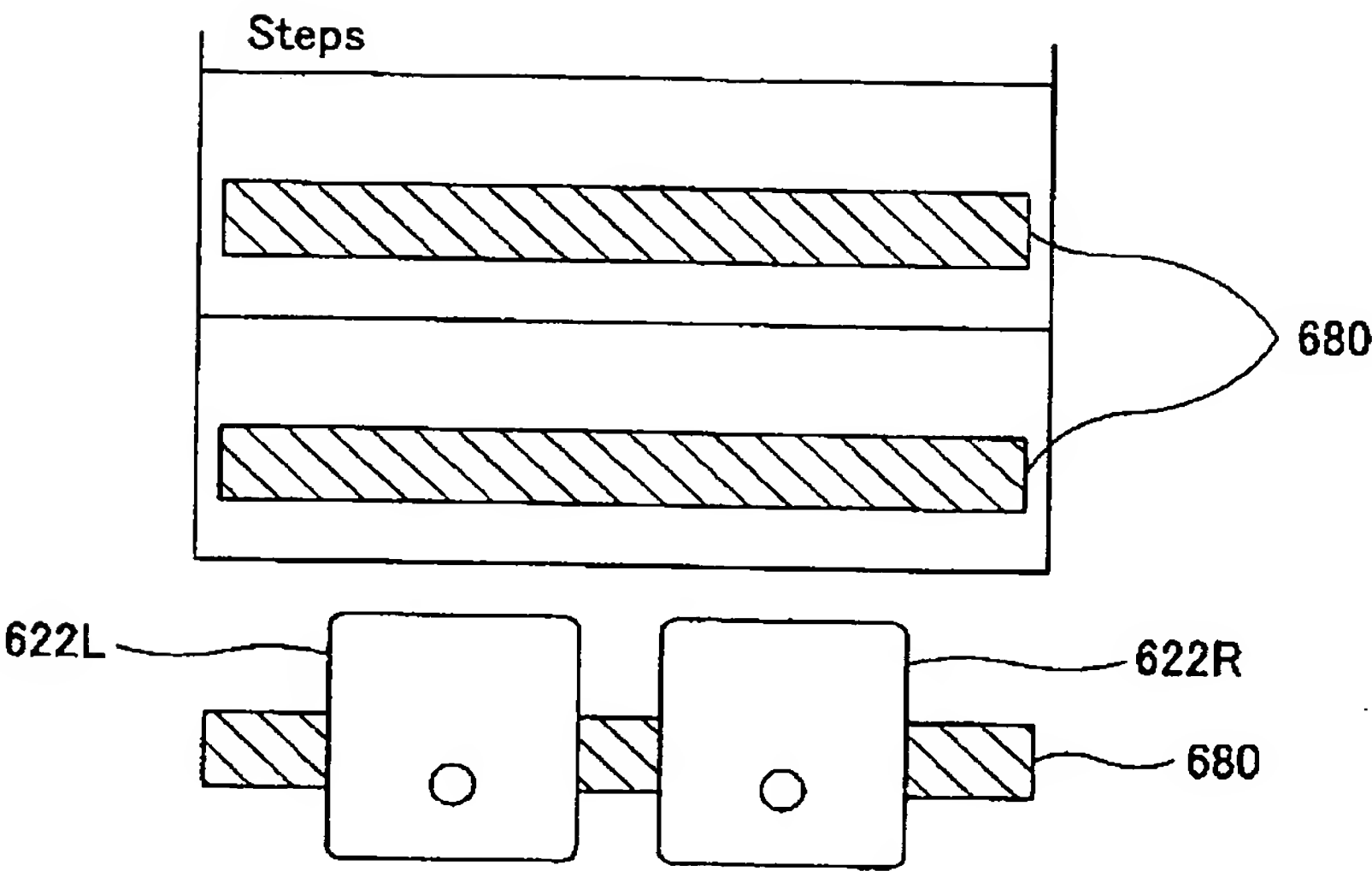
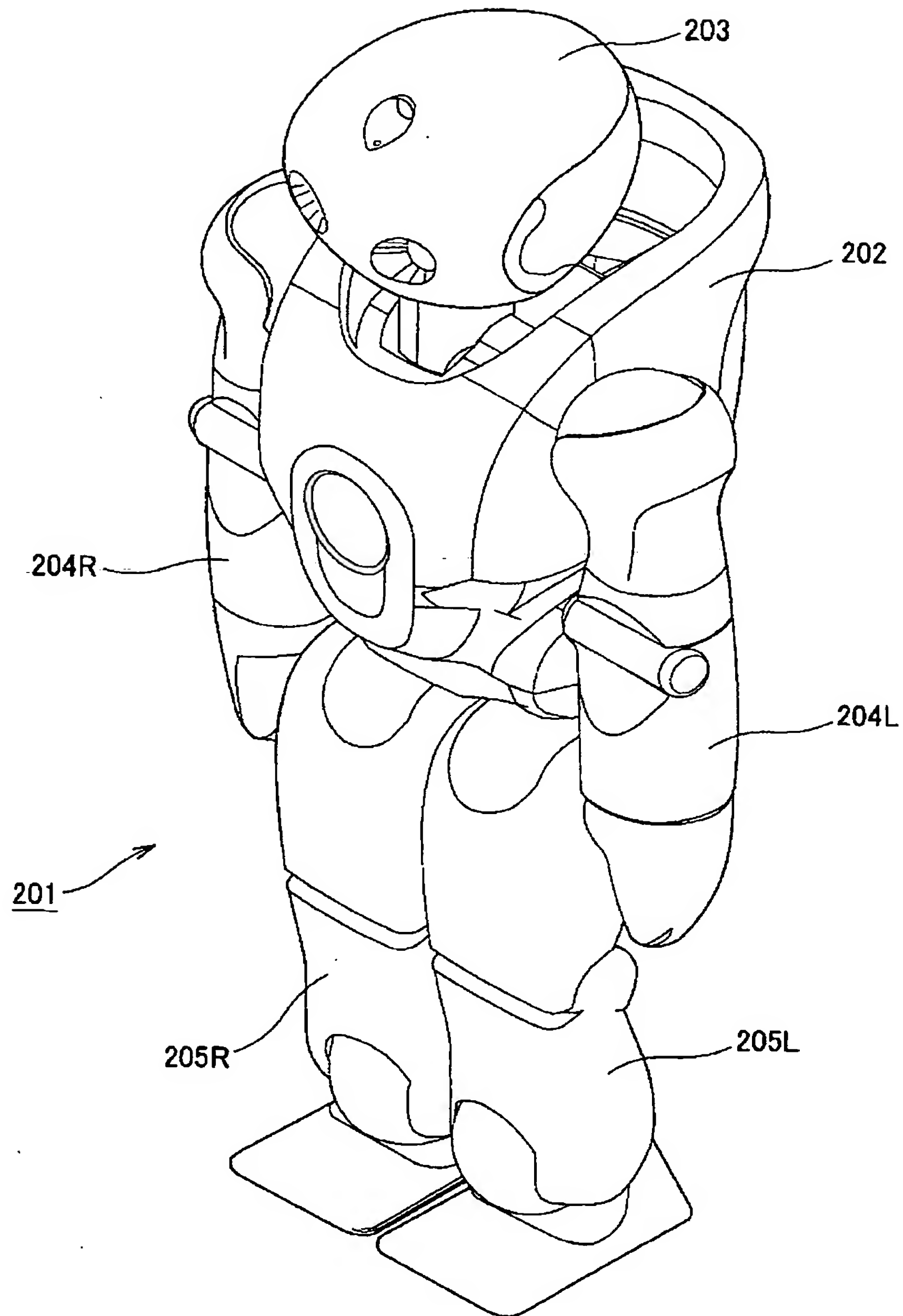


FIG.3B

[図4]

**FIG.4**

[図5]

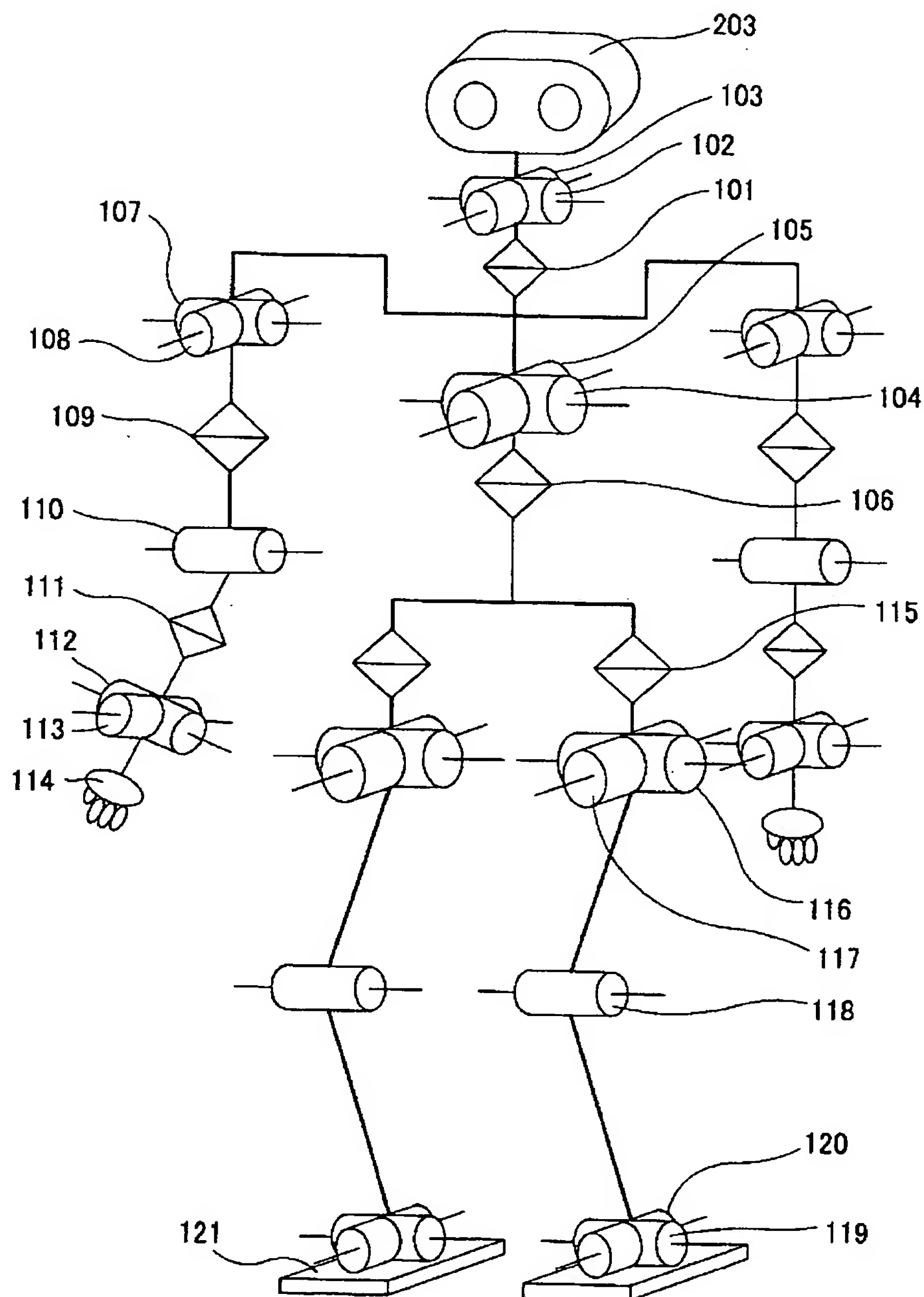


FIG.5

[図6]

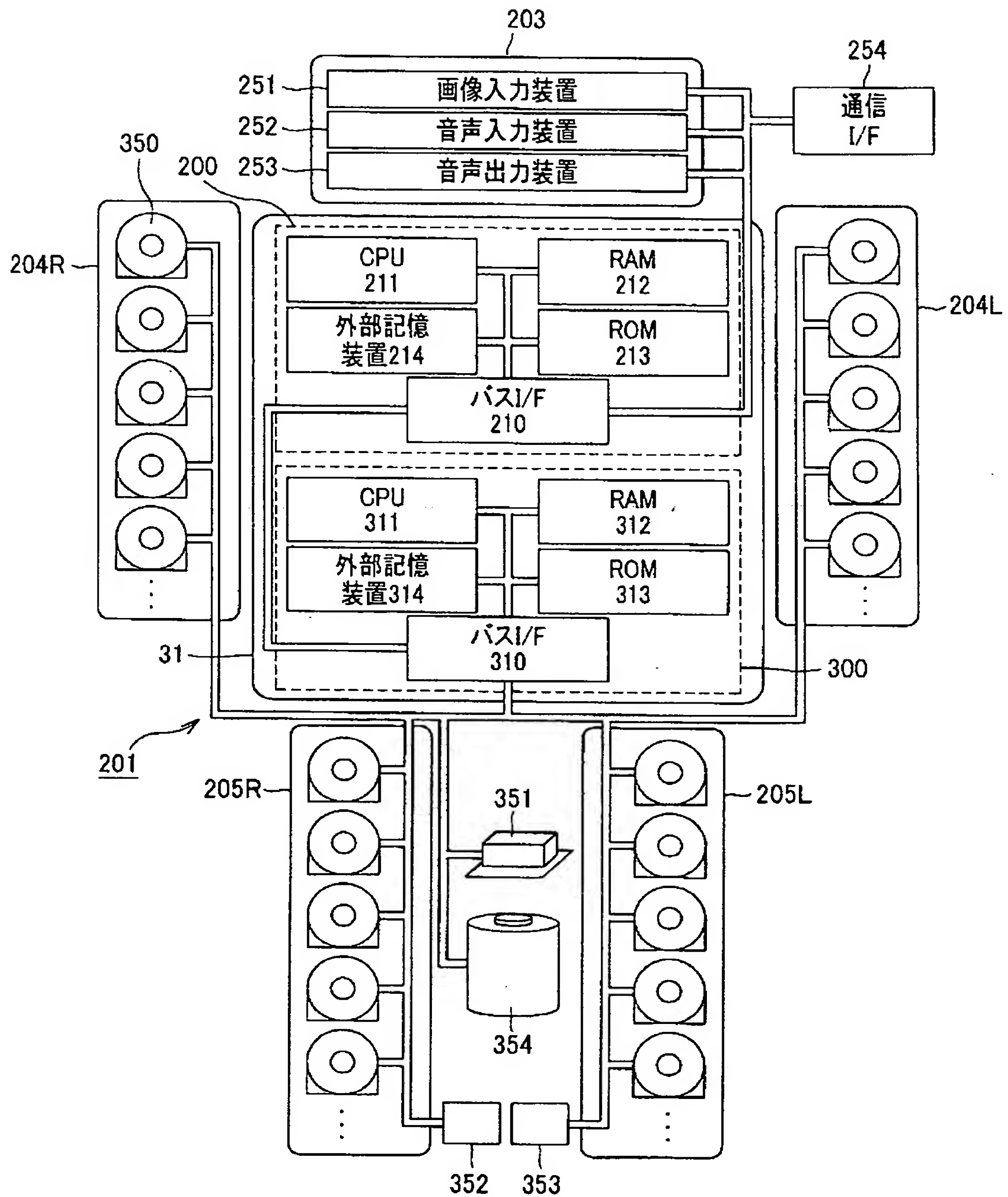


FIG. 6

[図7]

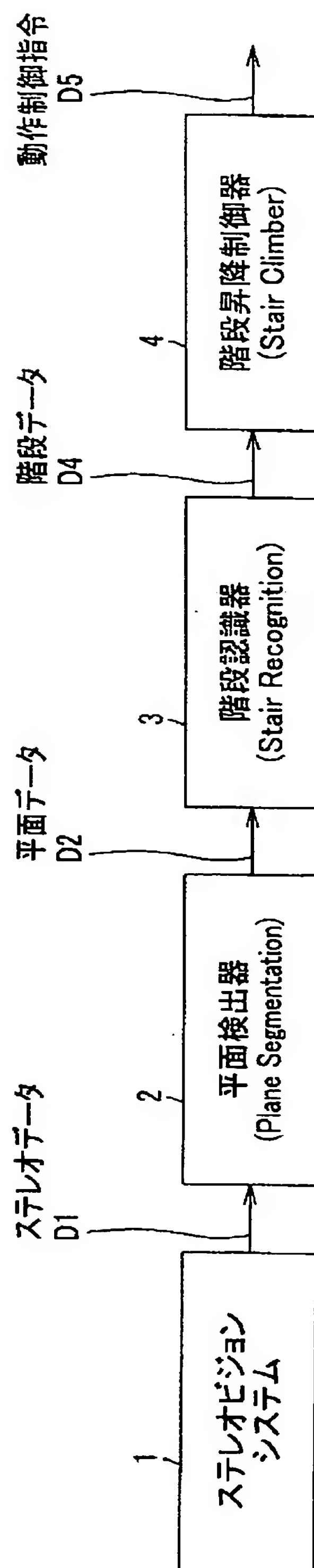


FIG.7



[図8]

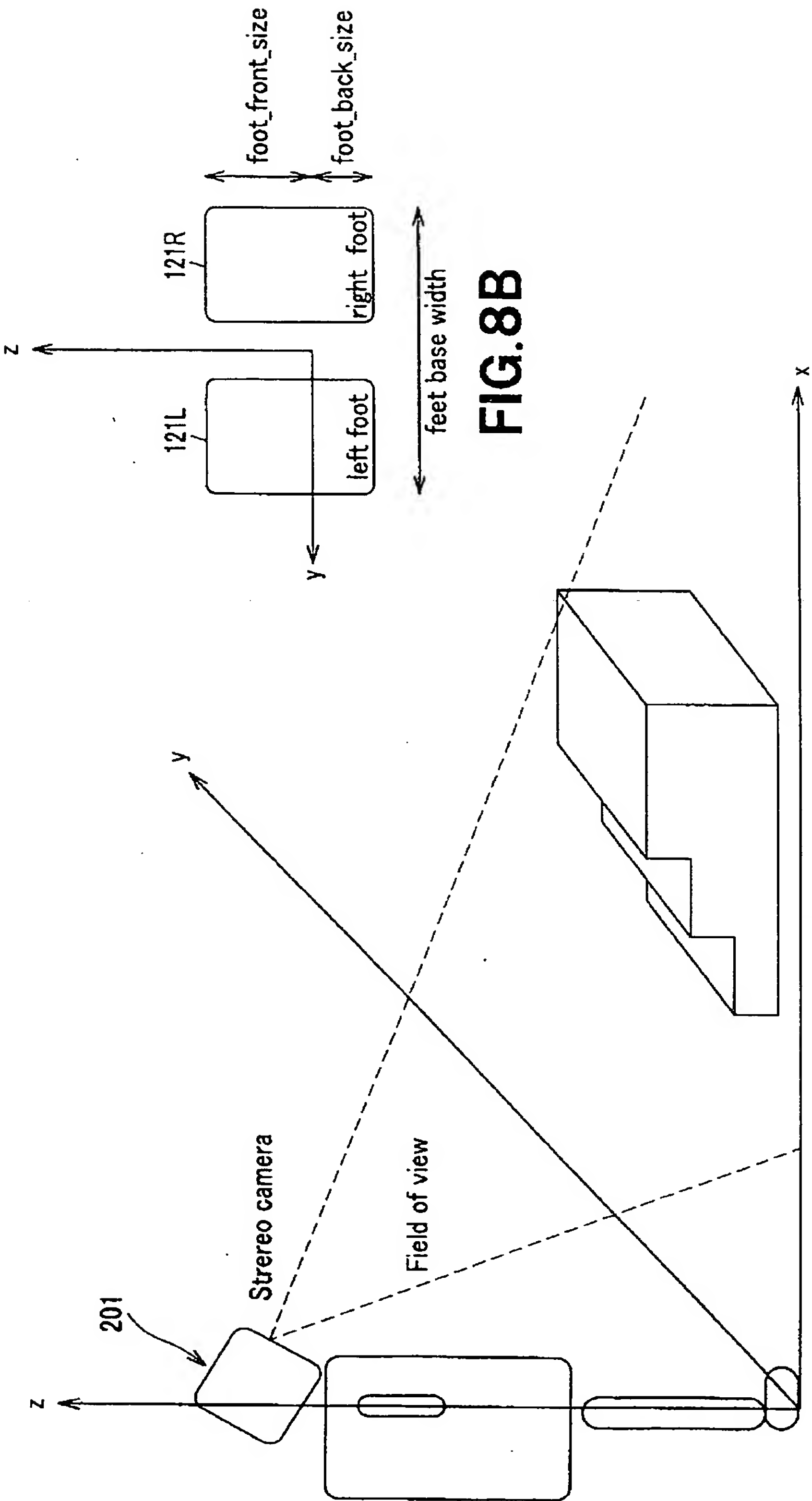
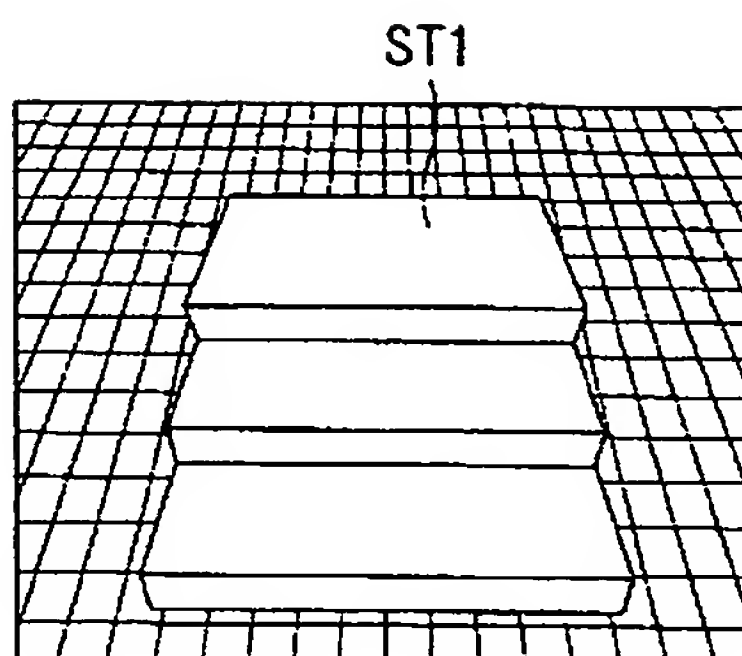


FIG.8B

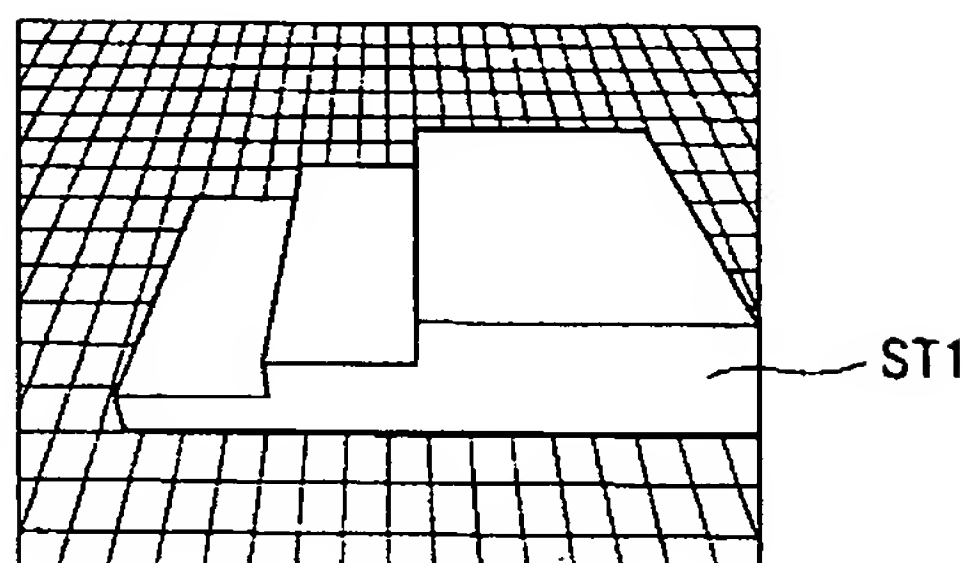
FIG.8A

[図9]

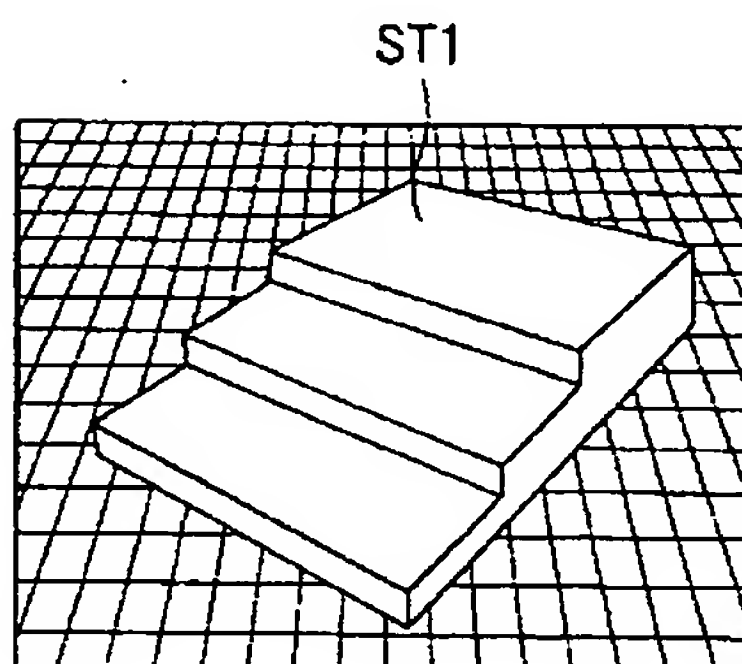
**FIG.9A**



**FIG.9B**



**FIG.9C**



4cm × 30cm × 10cm/21cm

[図10]

FIG.10A

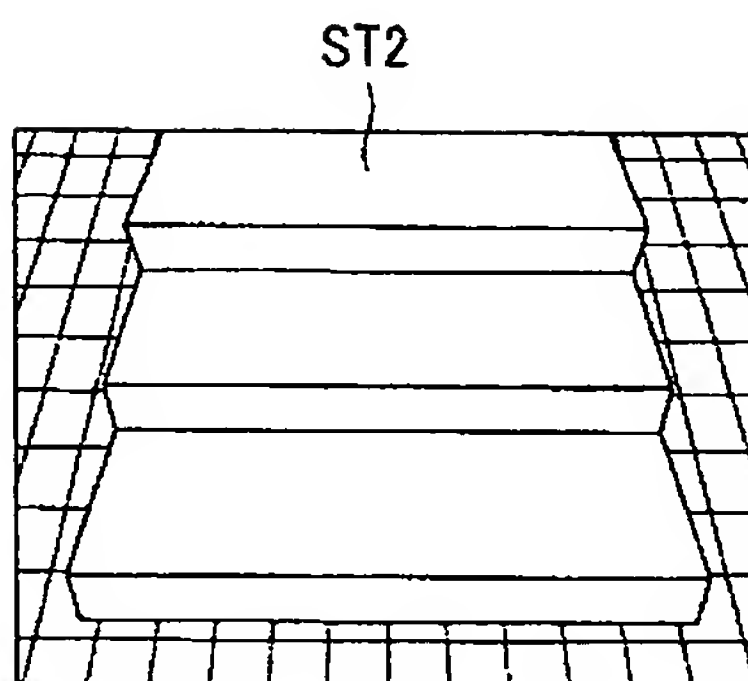


FIG.10B

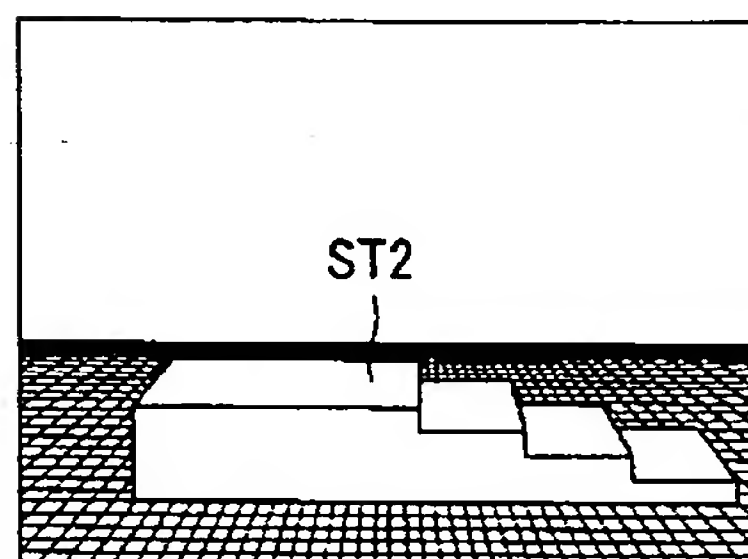
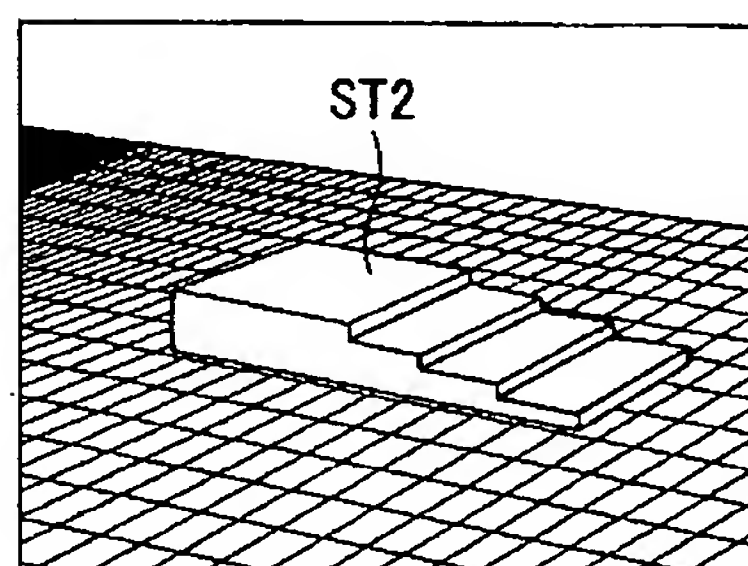


FIG.10C



3cm × 33cm × 12cm/32cm

[ 11 ]

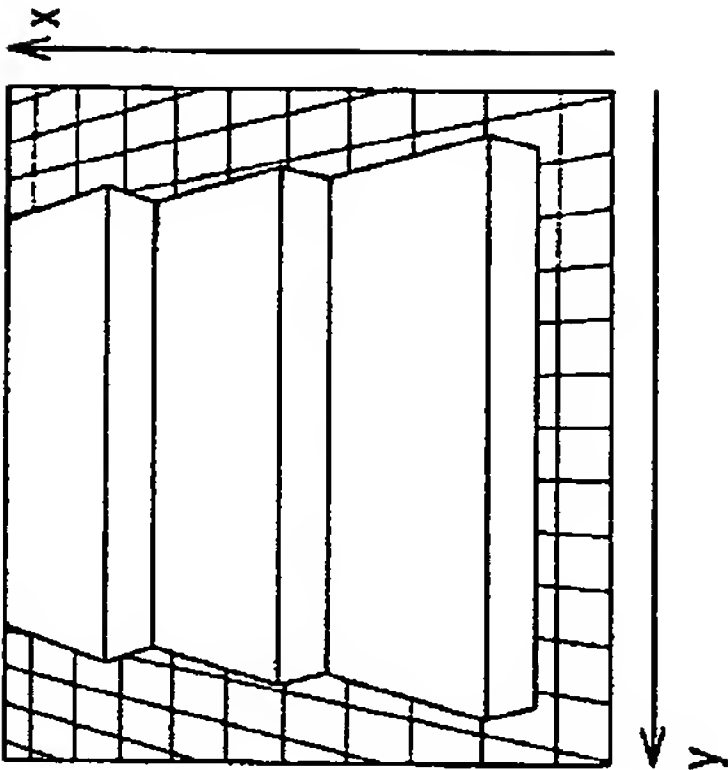


FIG. 11A

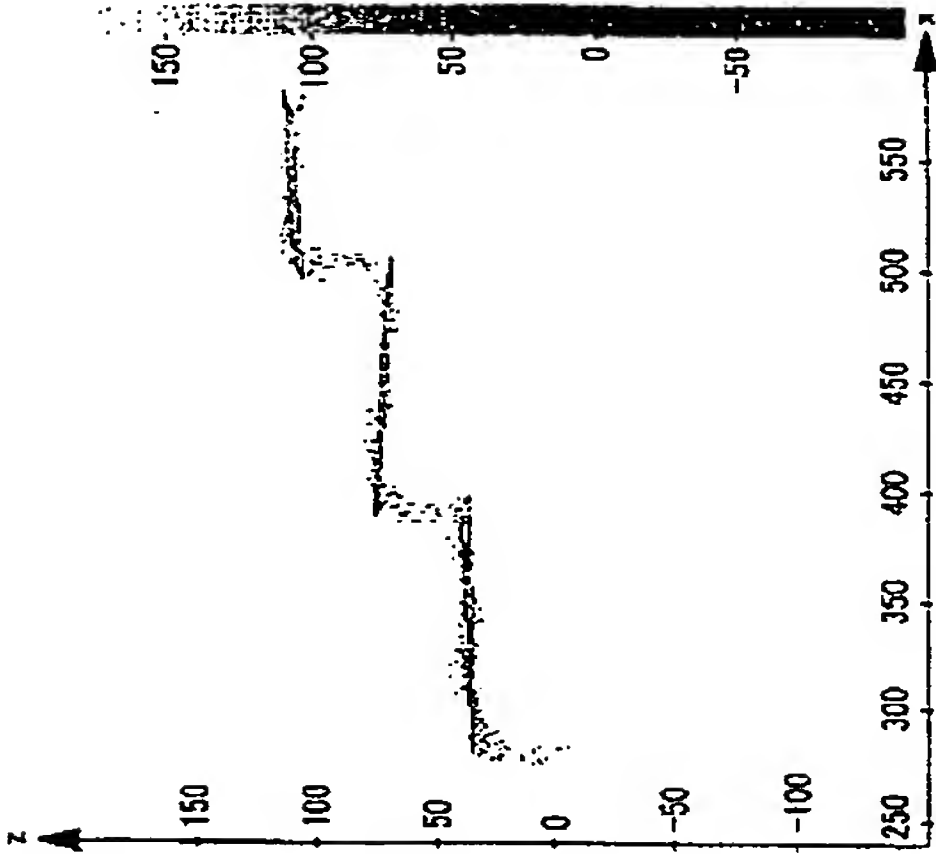


FIG. 11D

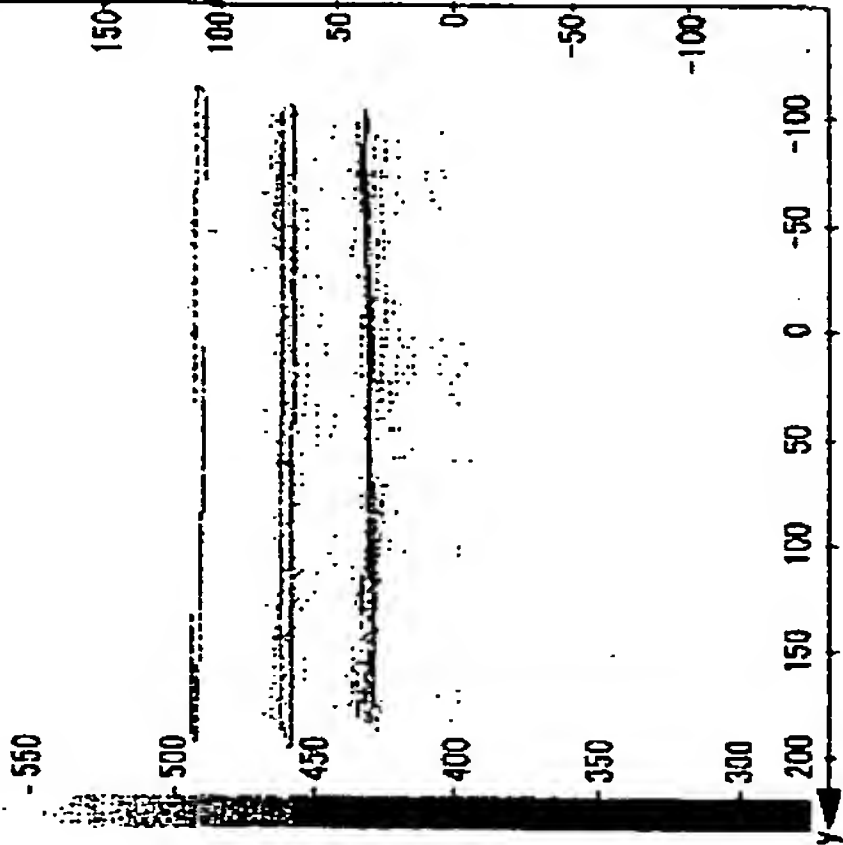


FIG. 11C

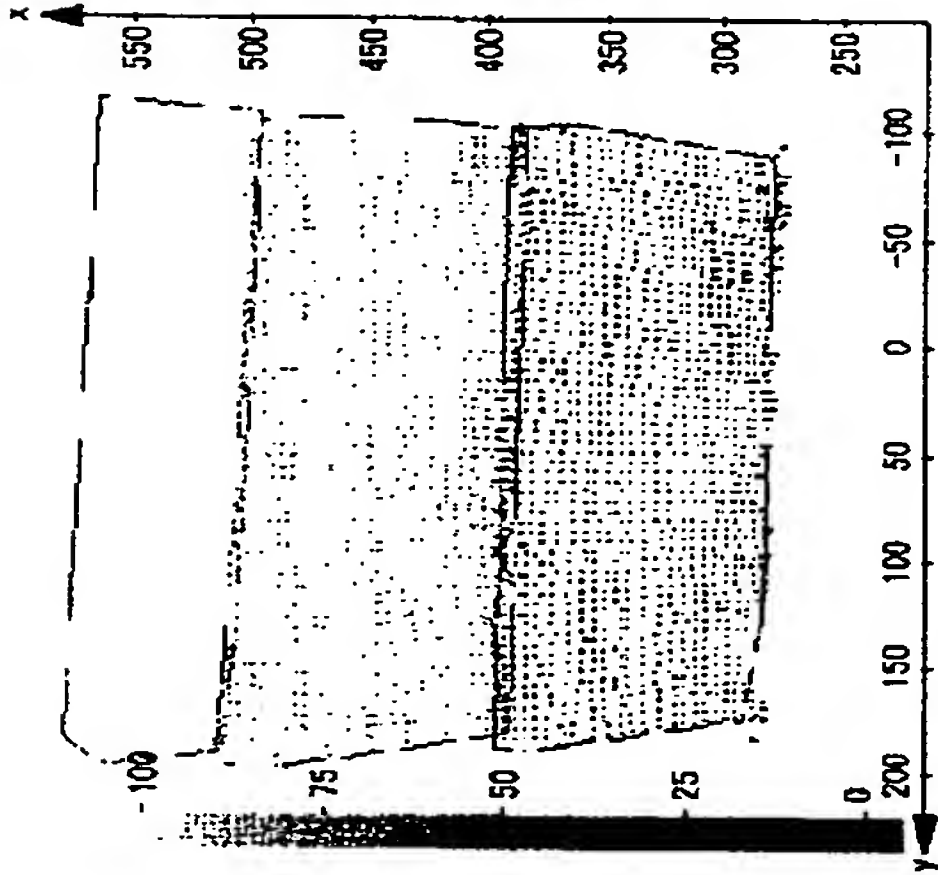


FIG. 11B

[図12]

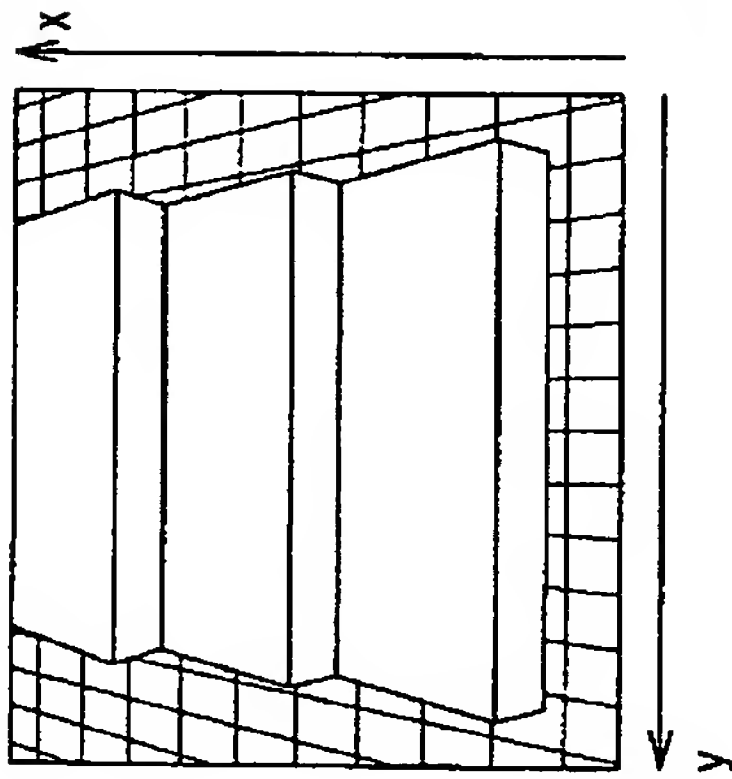


FIG. 12A

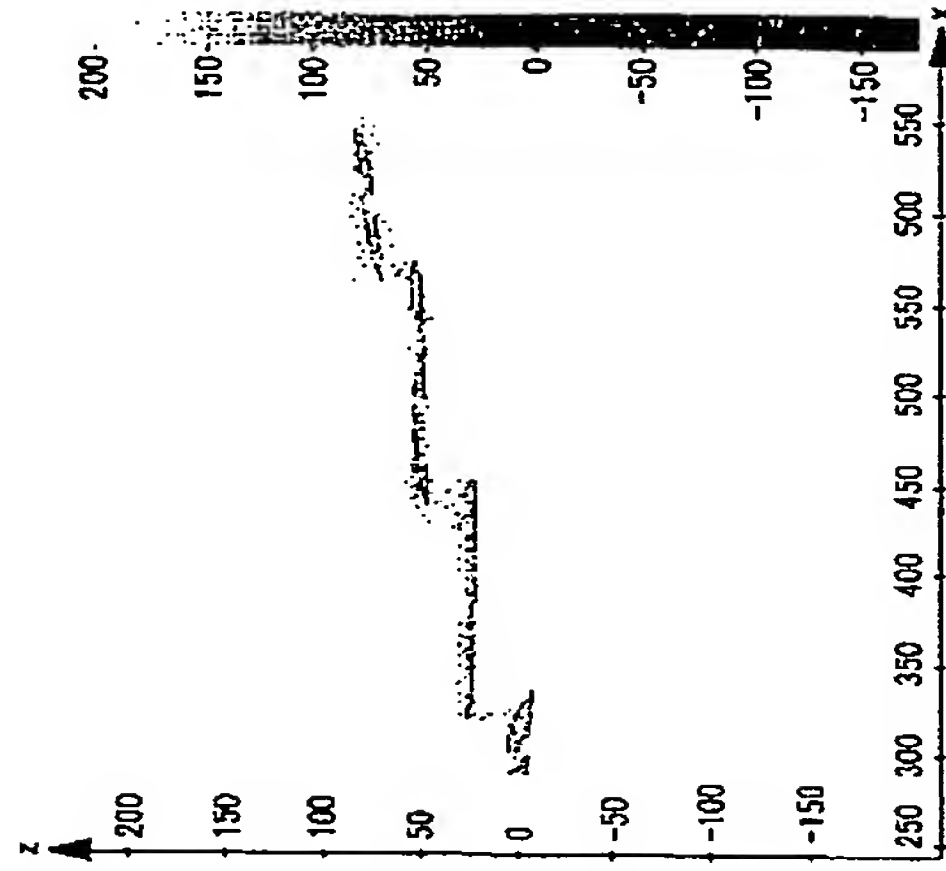


FIG. 12C

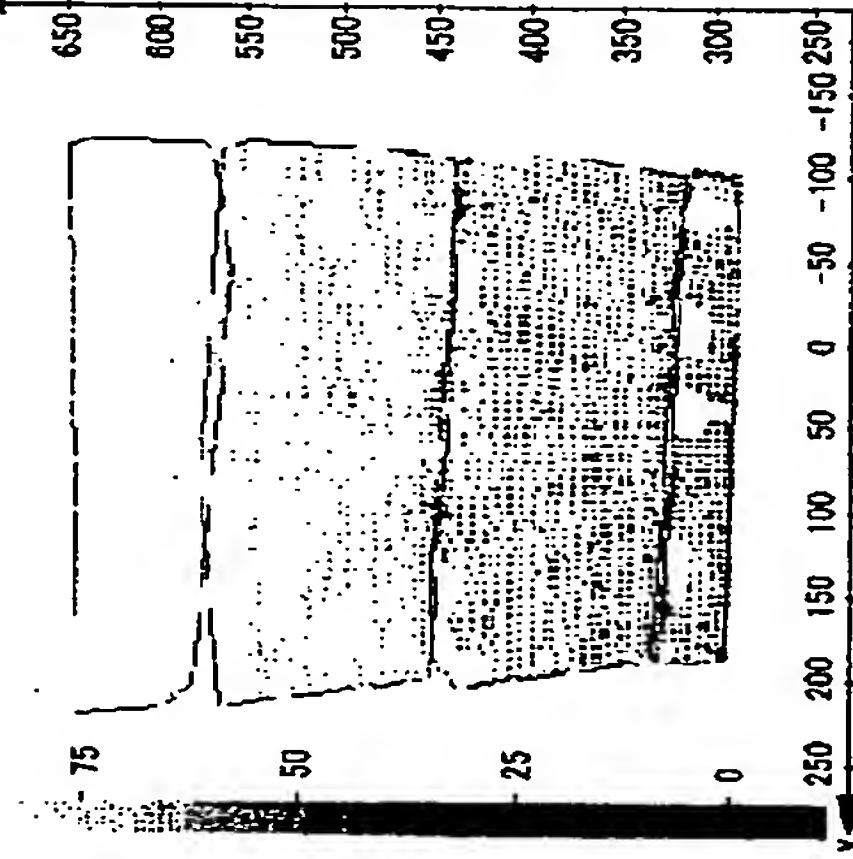


FIG. 12B

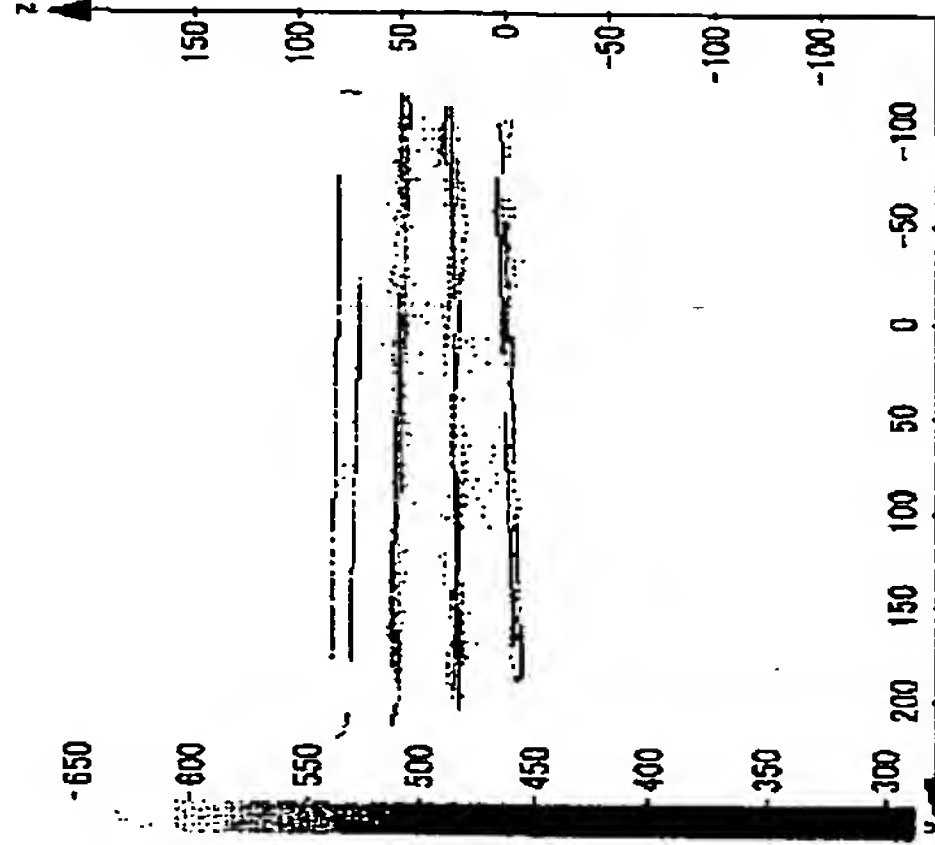


FIG. 12D

[図13]

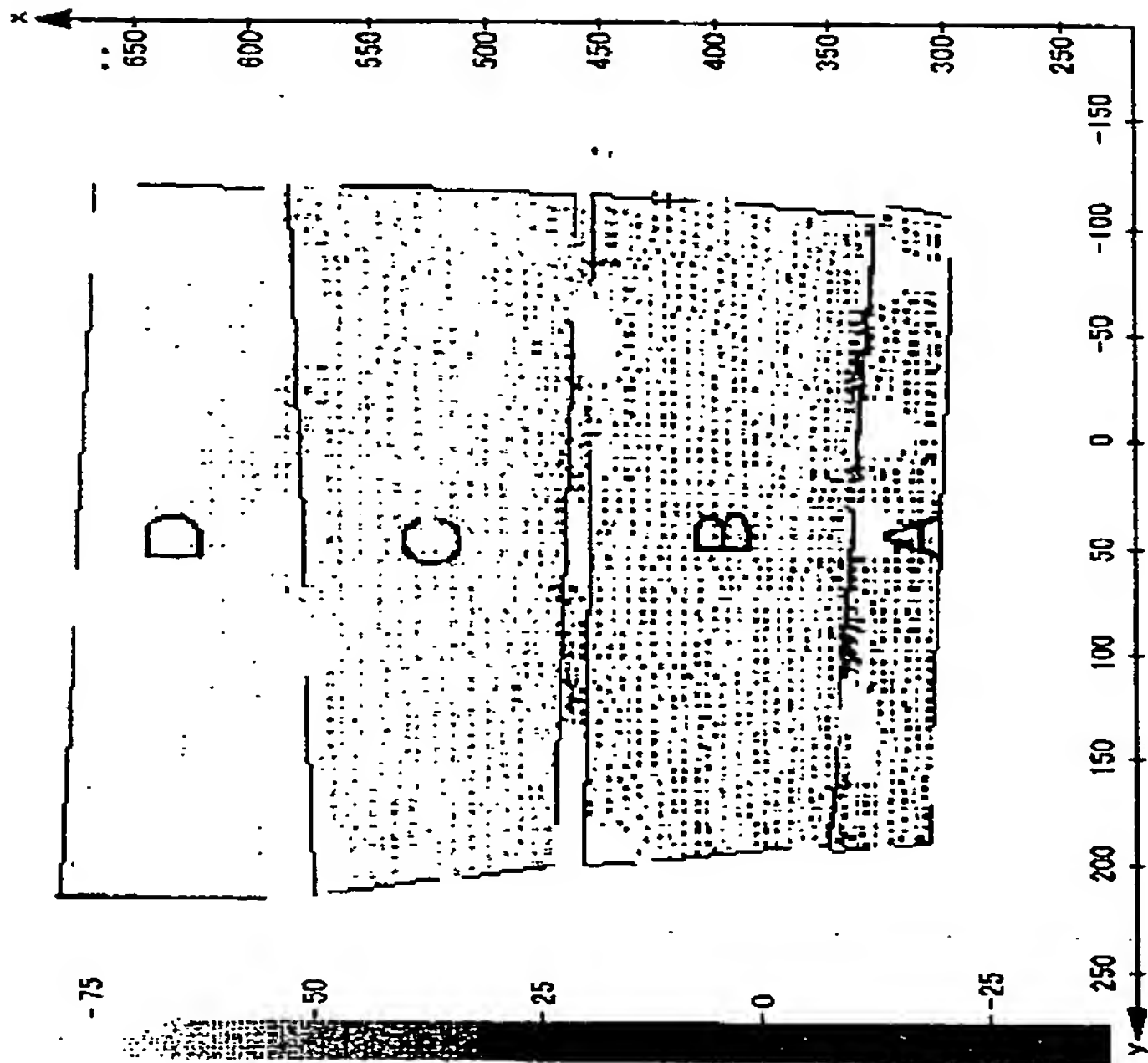


FIG.13B

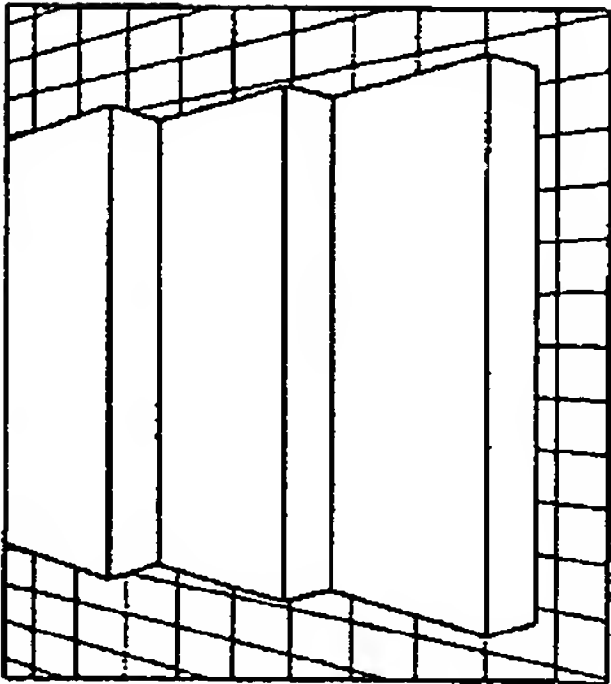


FIG.13A



[図14]

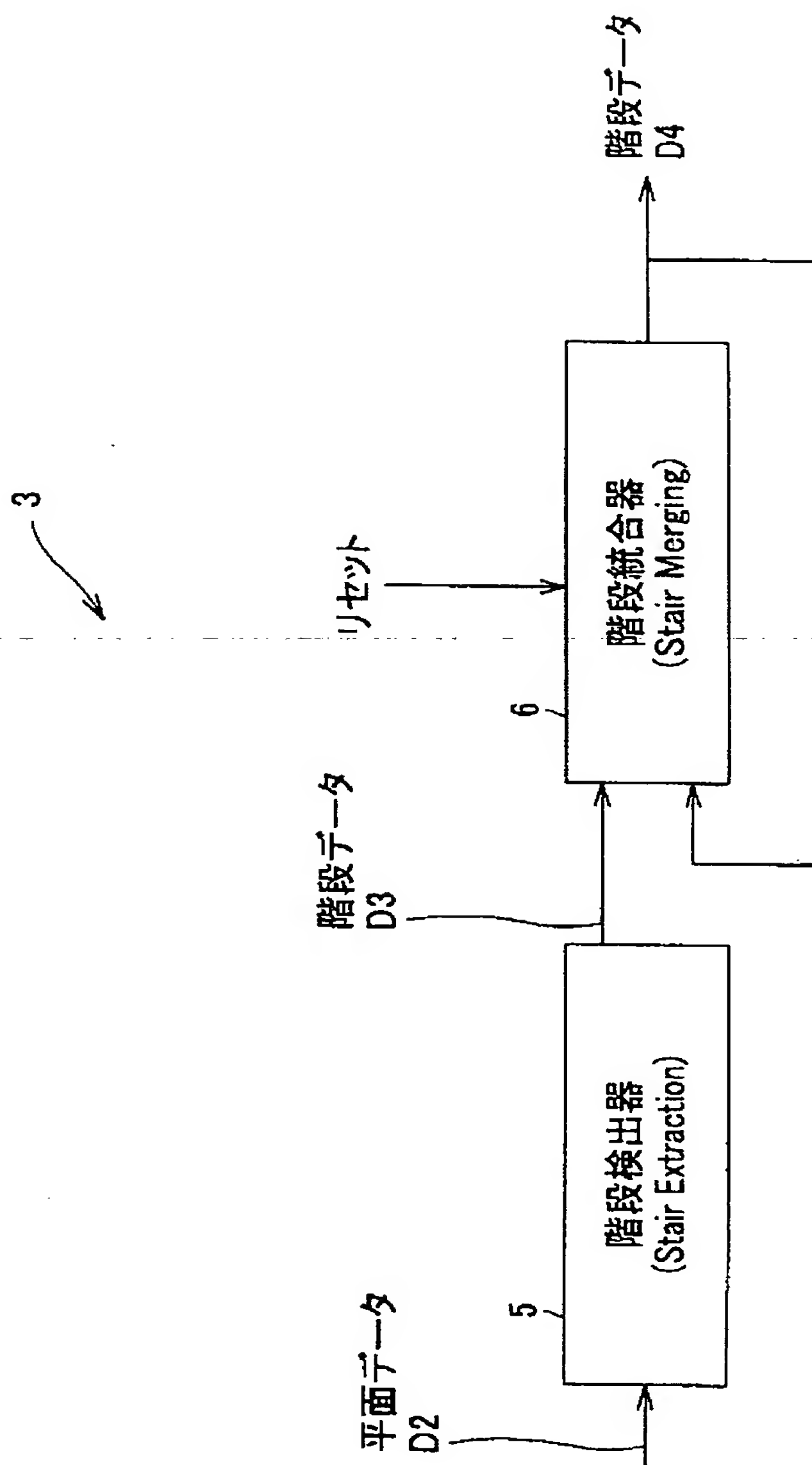


FIG.14

[図15]

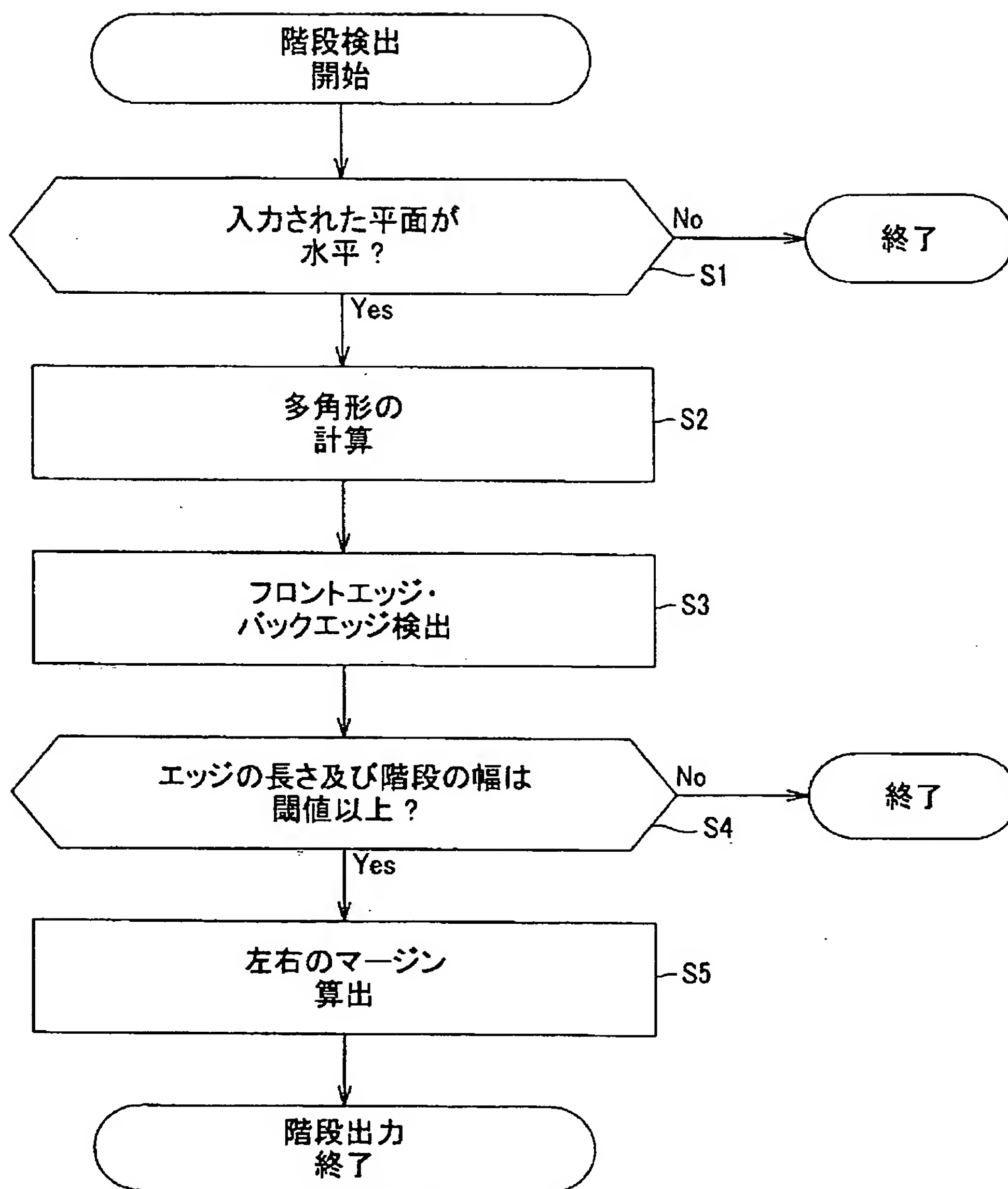


FIG.15

[図16]

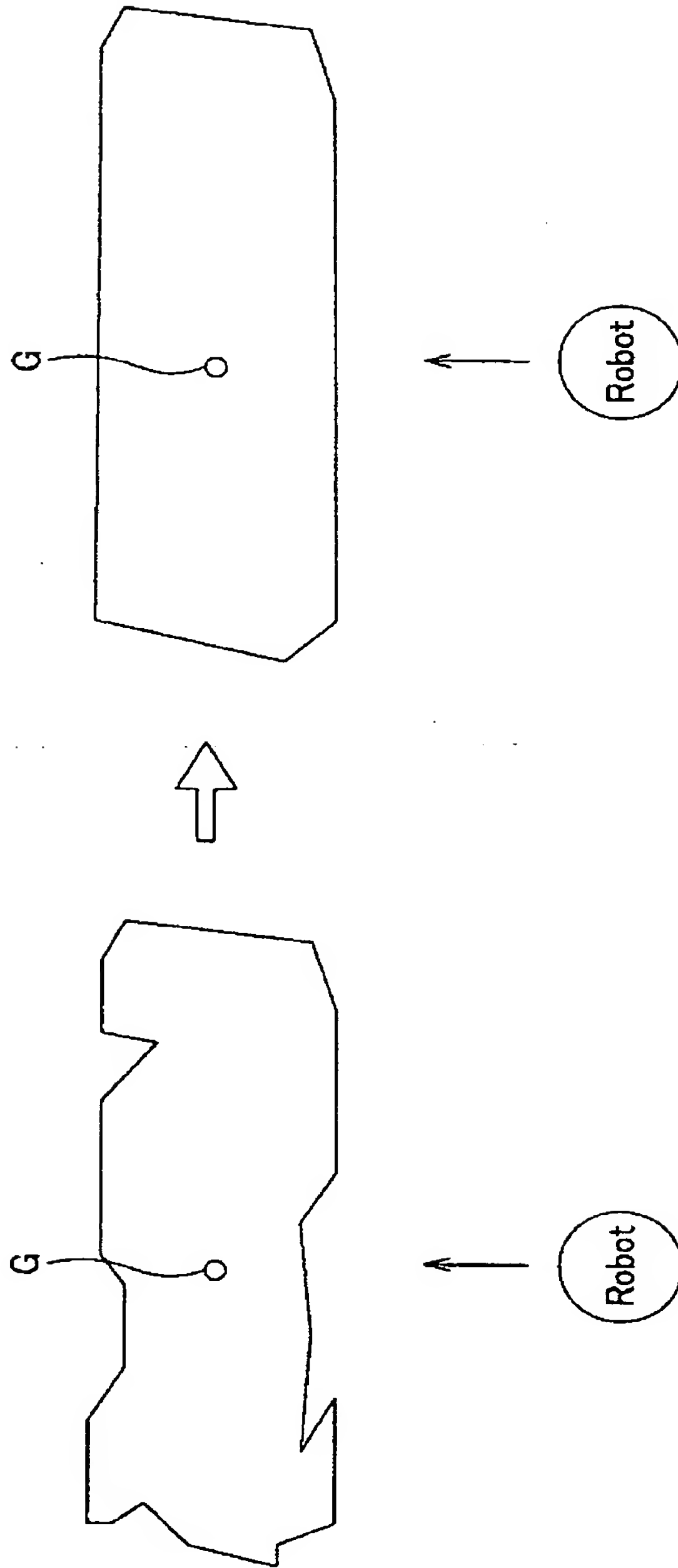


FIG. 16B

FIG. 16A

[図17]

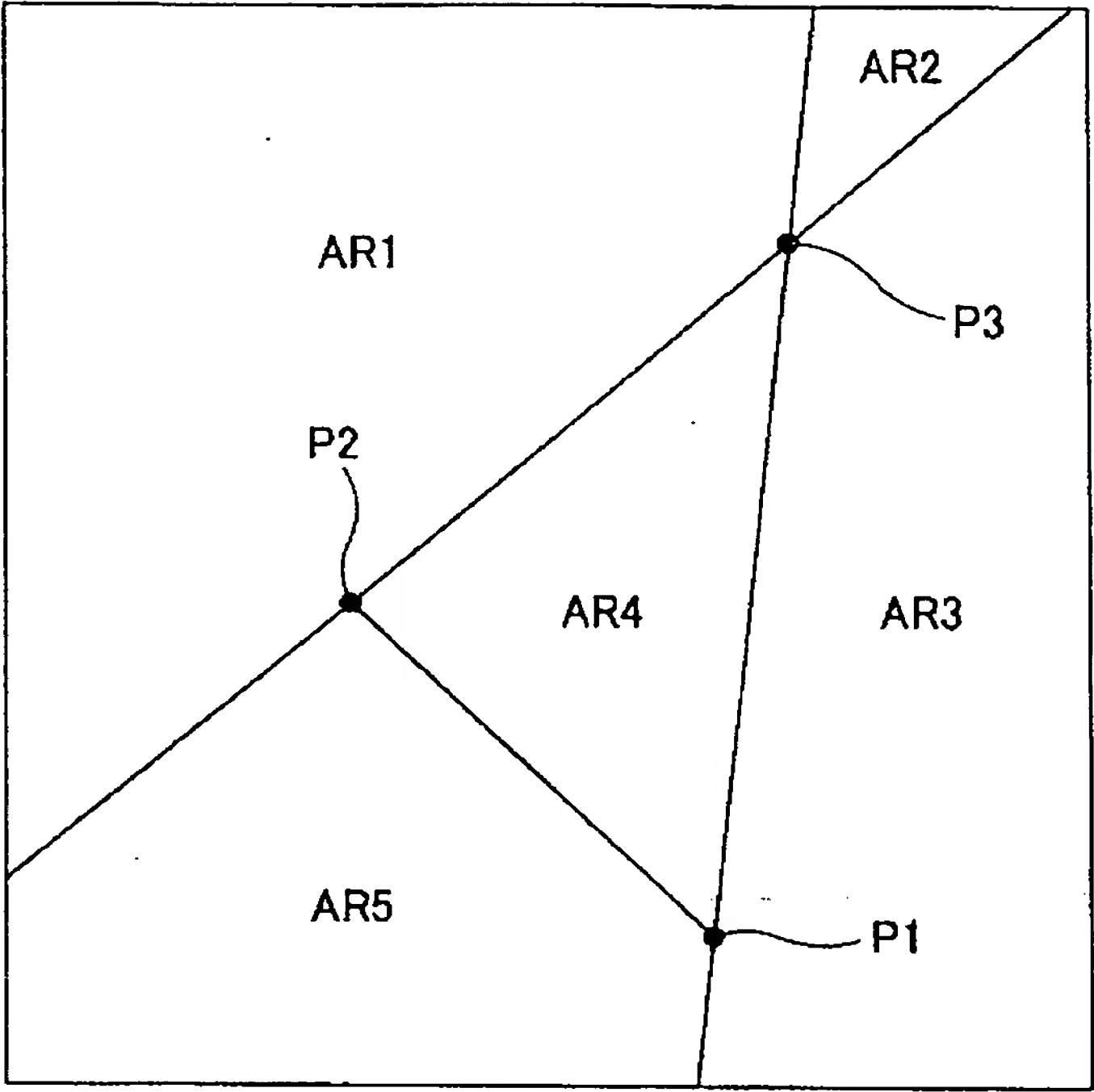
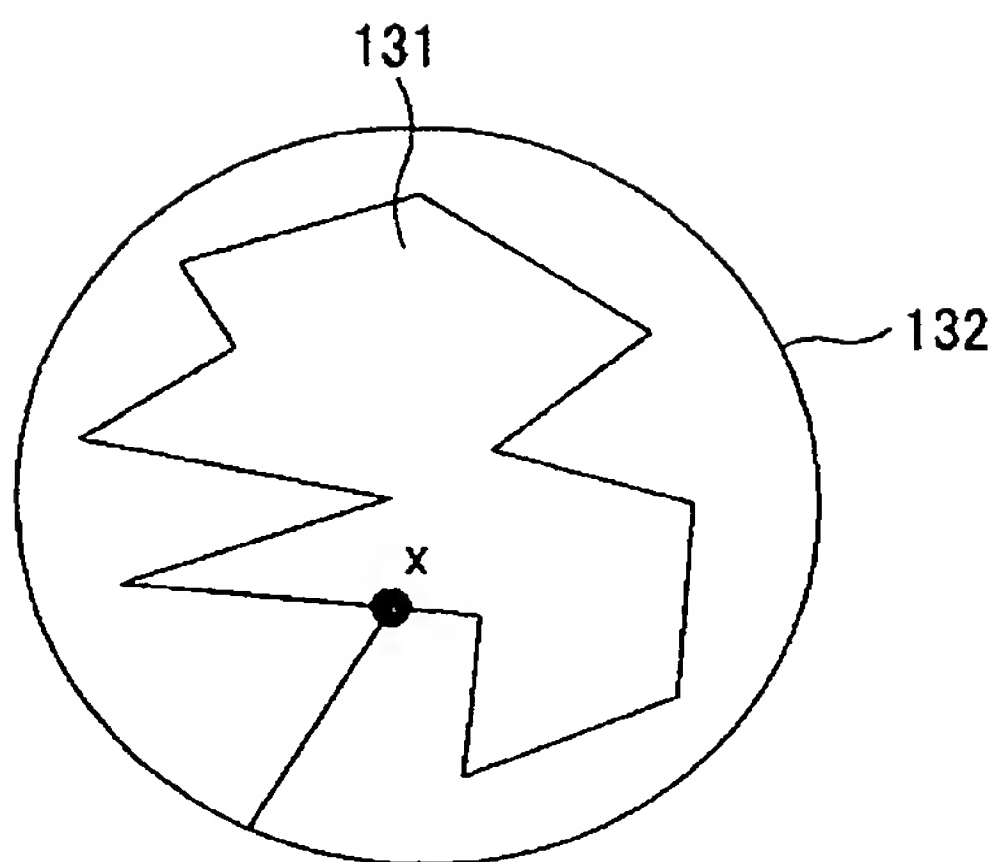
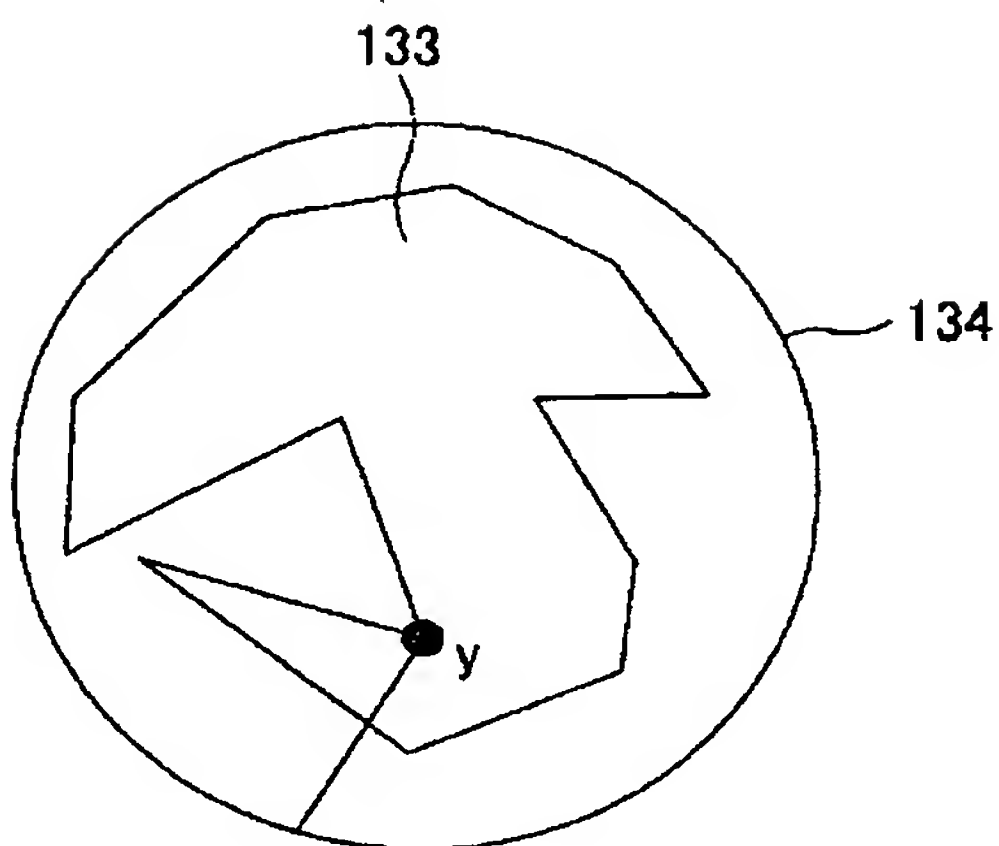


FIG.17

[図18]



**FIG. 18A**



**FIG. 18B**

[図19]

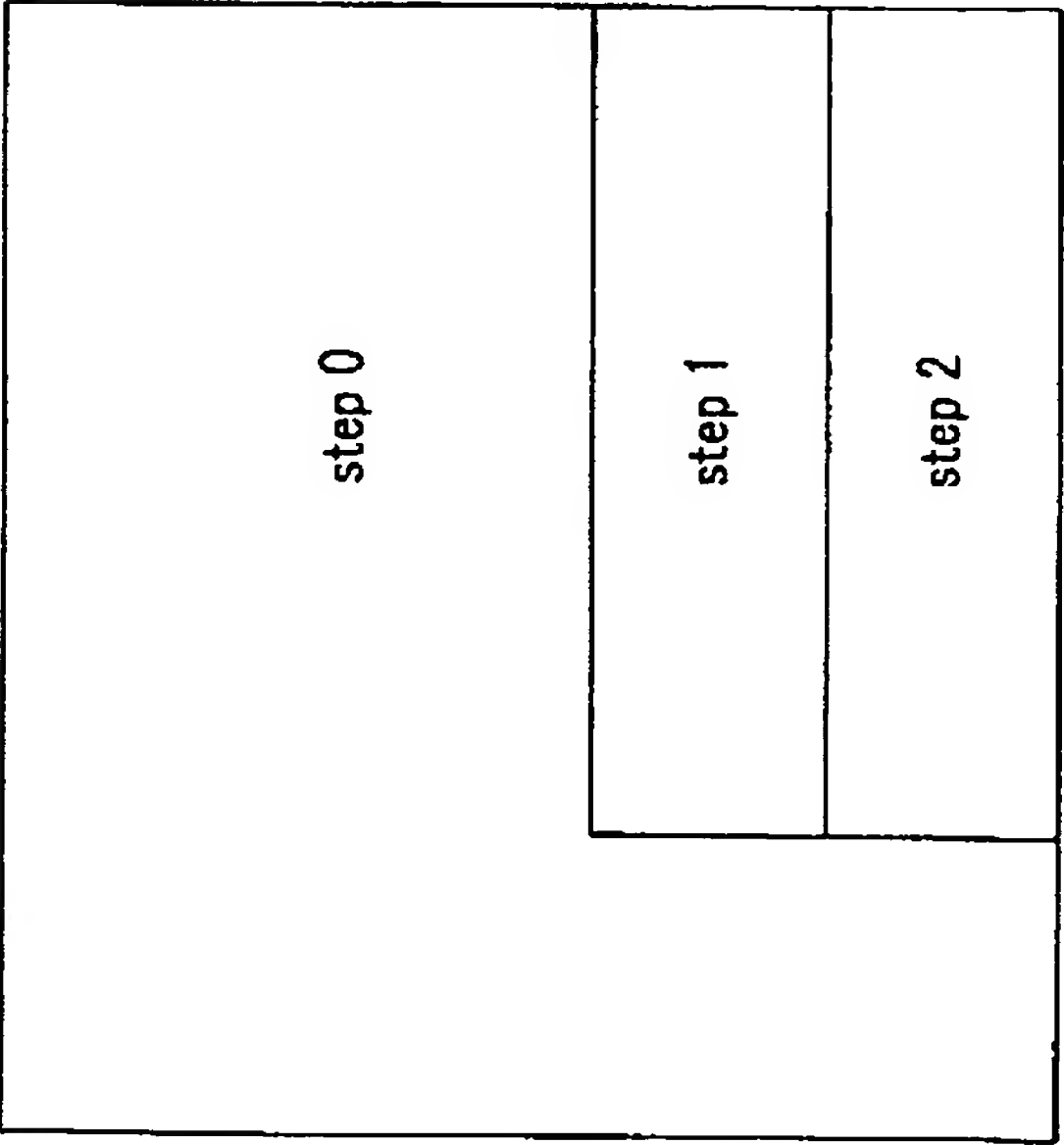
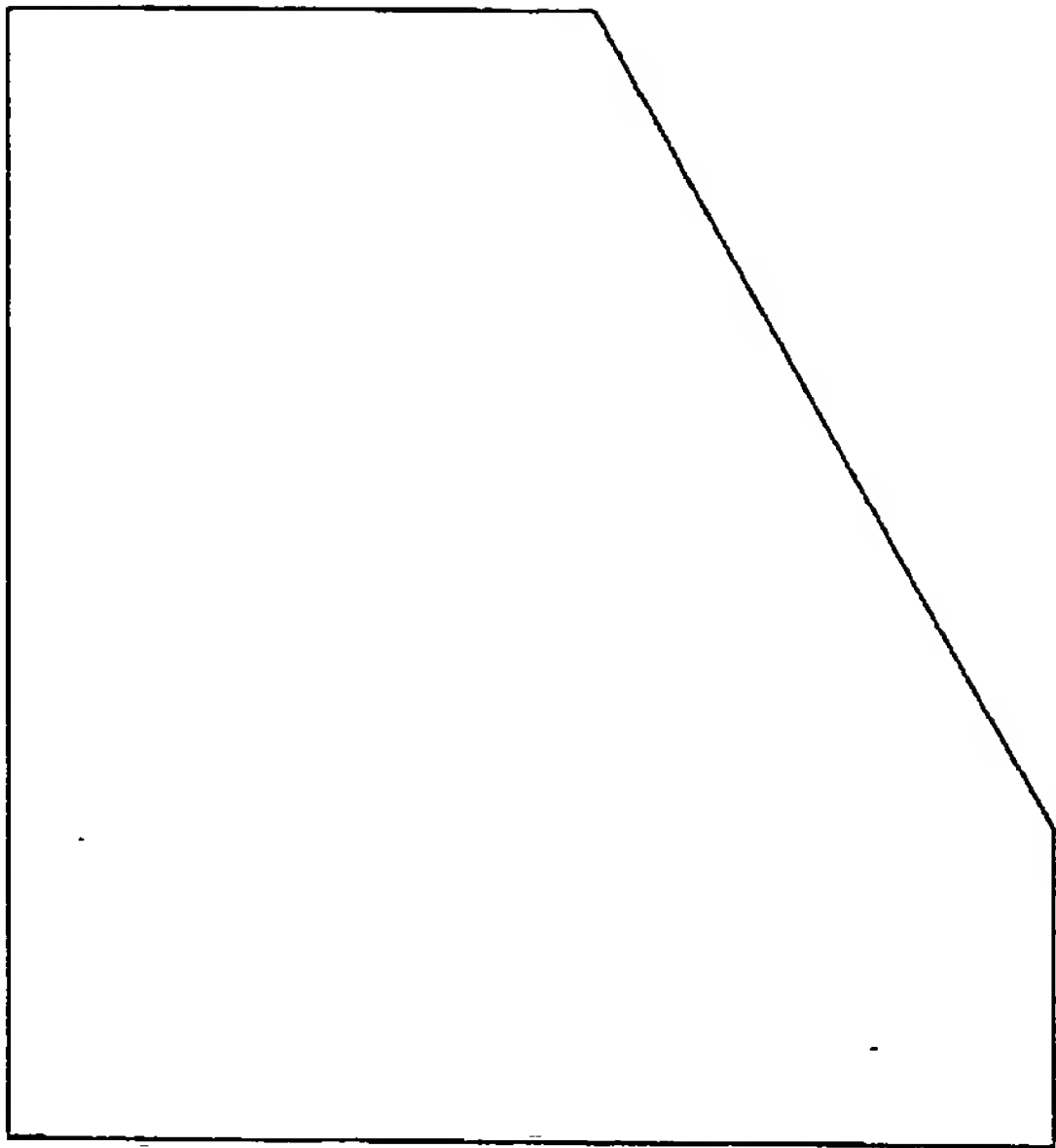


FIG.19B

FIG.19A



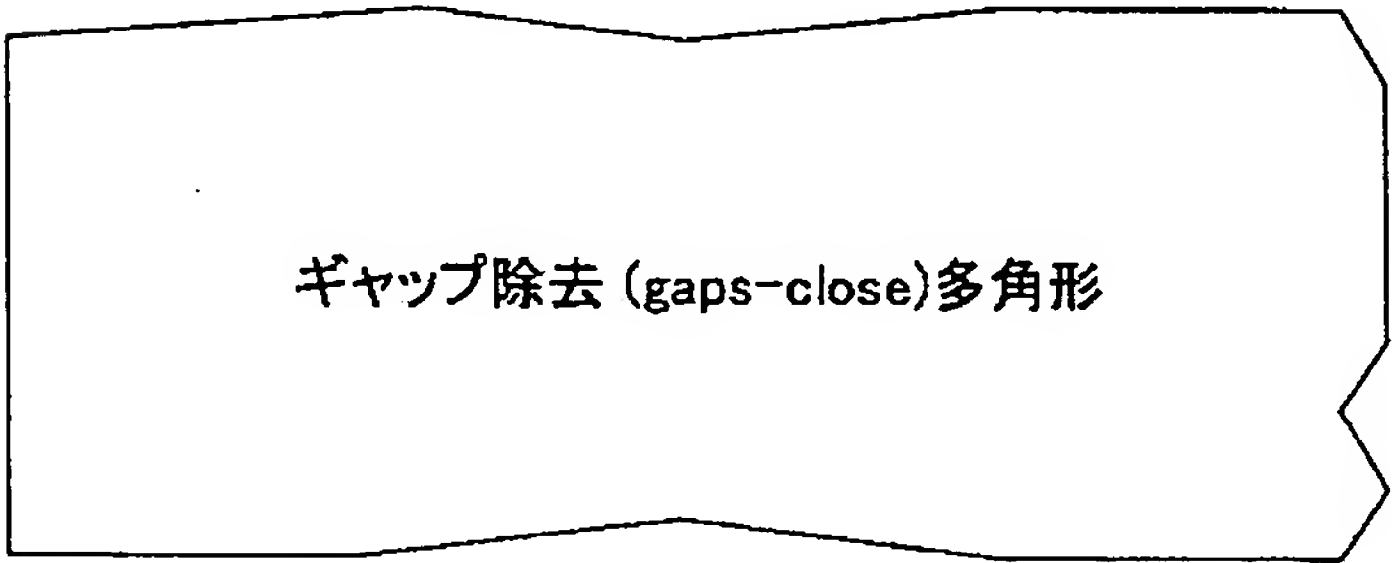
[図20]

FIG.20A



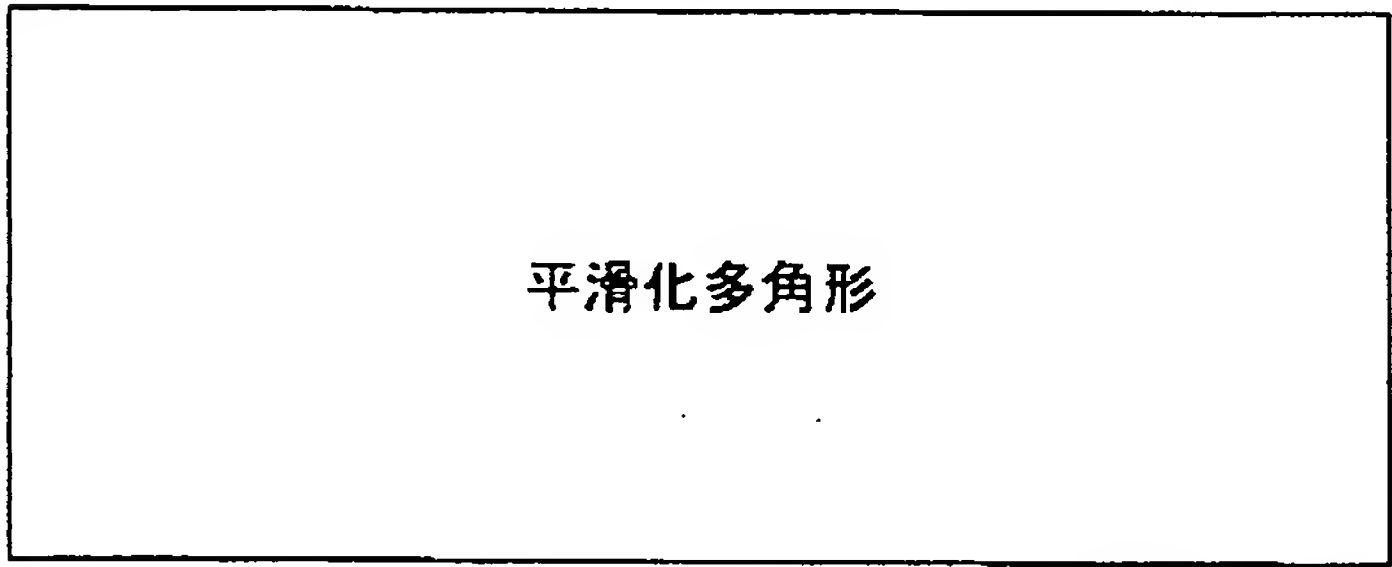
↓ close gaps

FIG.20B



↓ fit line segments

FIG.20C



[図21]

**Close gaps :**

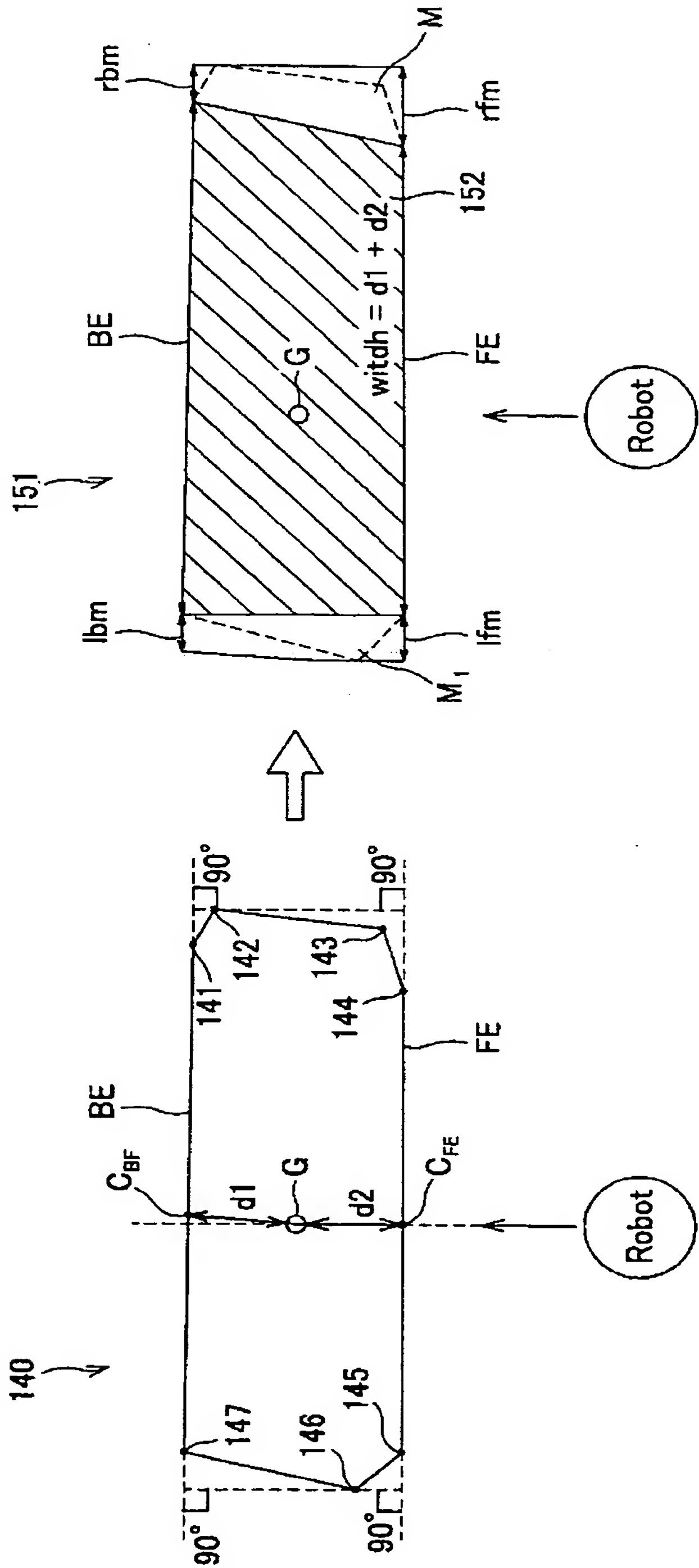
```
for each triple of consecutive points do
  if distance first to last point  $< d_{\text{gap}}$  then
    remove middle point
  end
until no more changes
```

**Fit line segments :**

```
for each triple of consecutive points do
  compute least-squares line fit with error
end
for each fitted line, sorted by error do
  if error  $< e_{\text{fit}}$  then
    remove middle point
    update first and last point
    re-compute fitted lines for points
      preceding and following line
  end
until no more changes
```

**FIG.21**

[図22]



[図23]

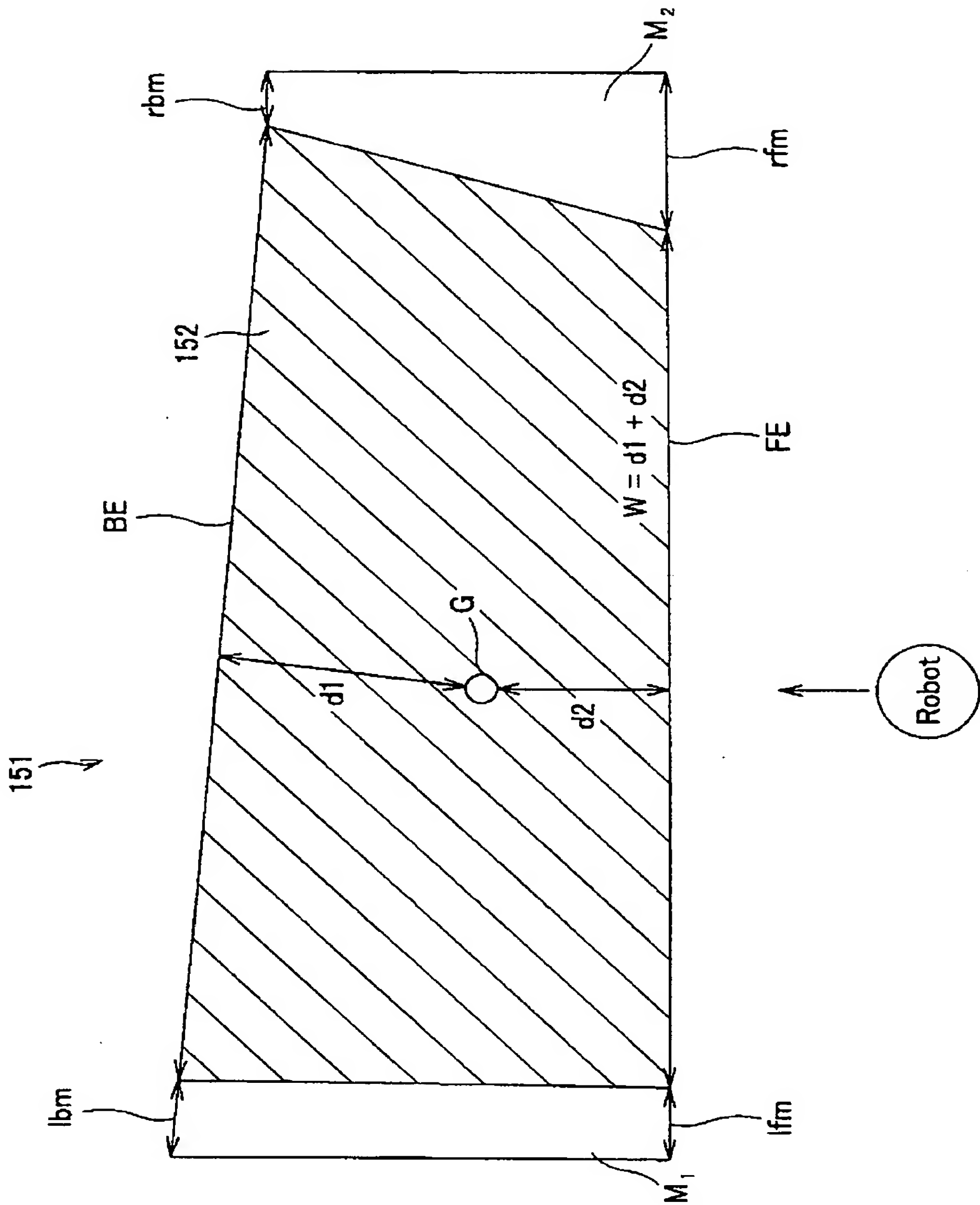


FIG.23

[図24]

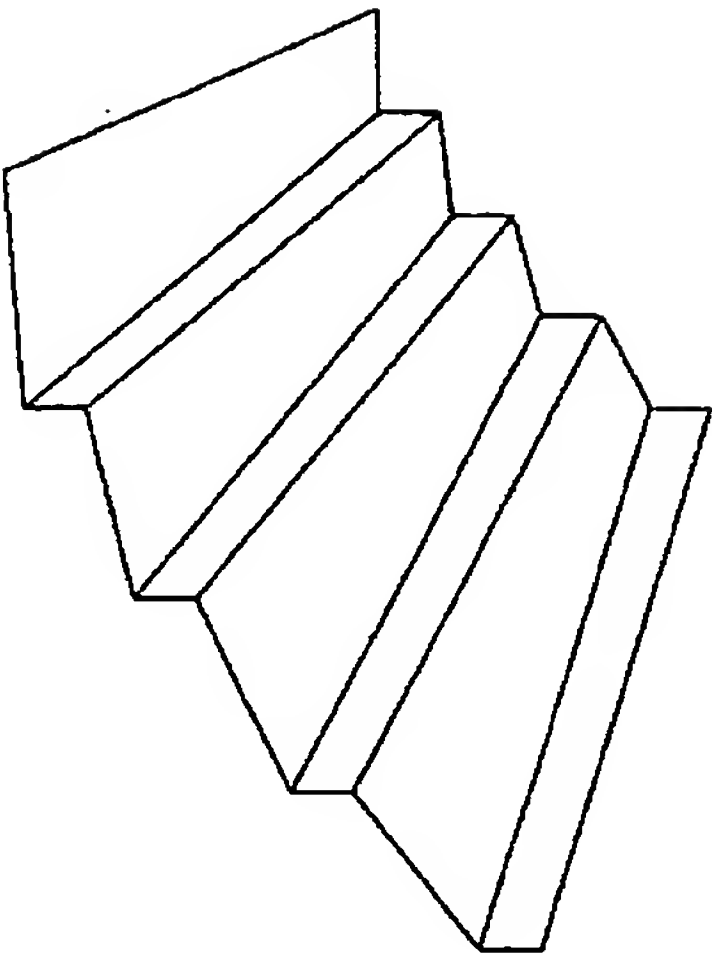


FIG.24B

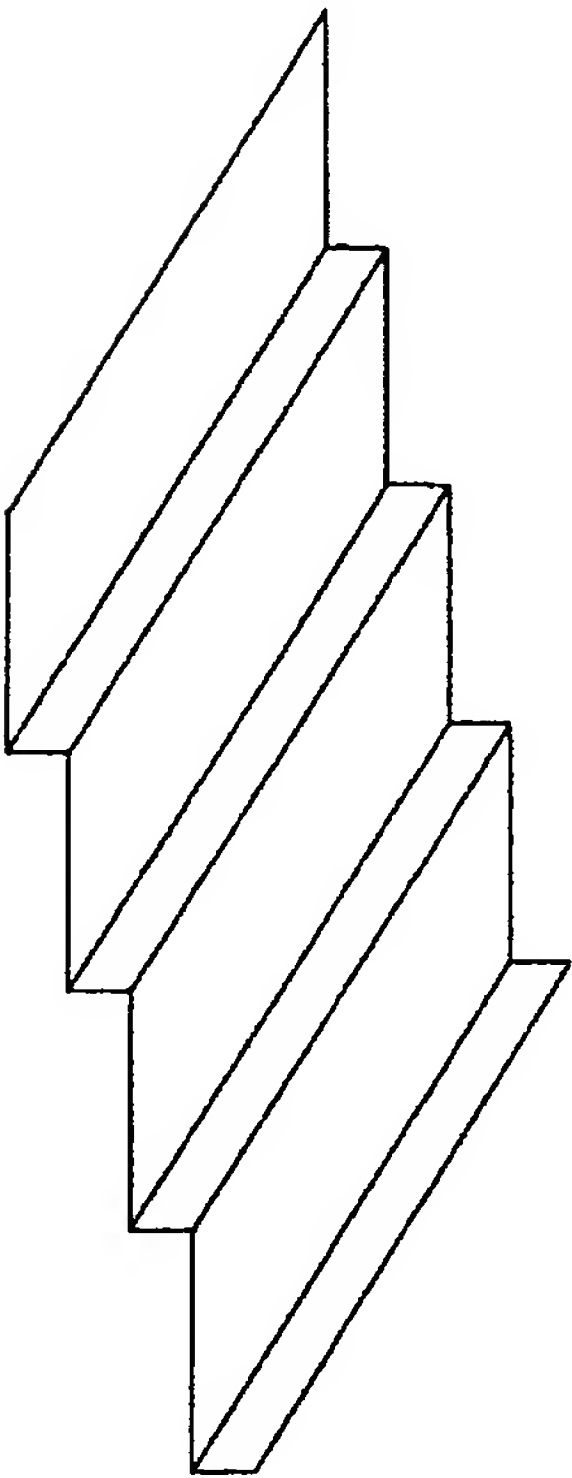


FIG.24A

[図25]

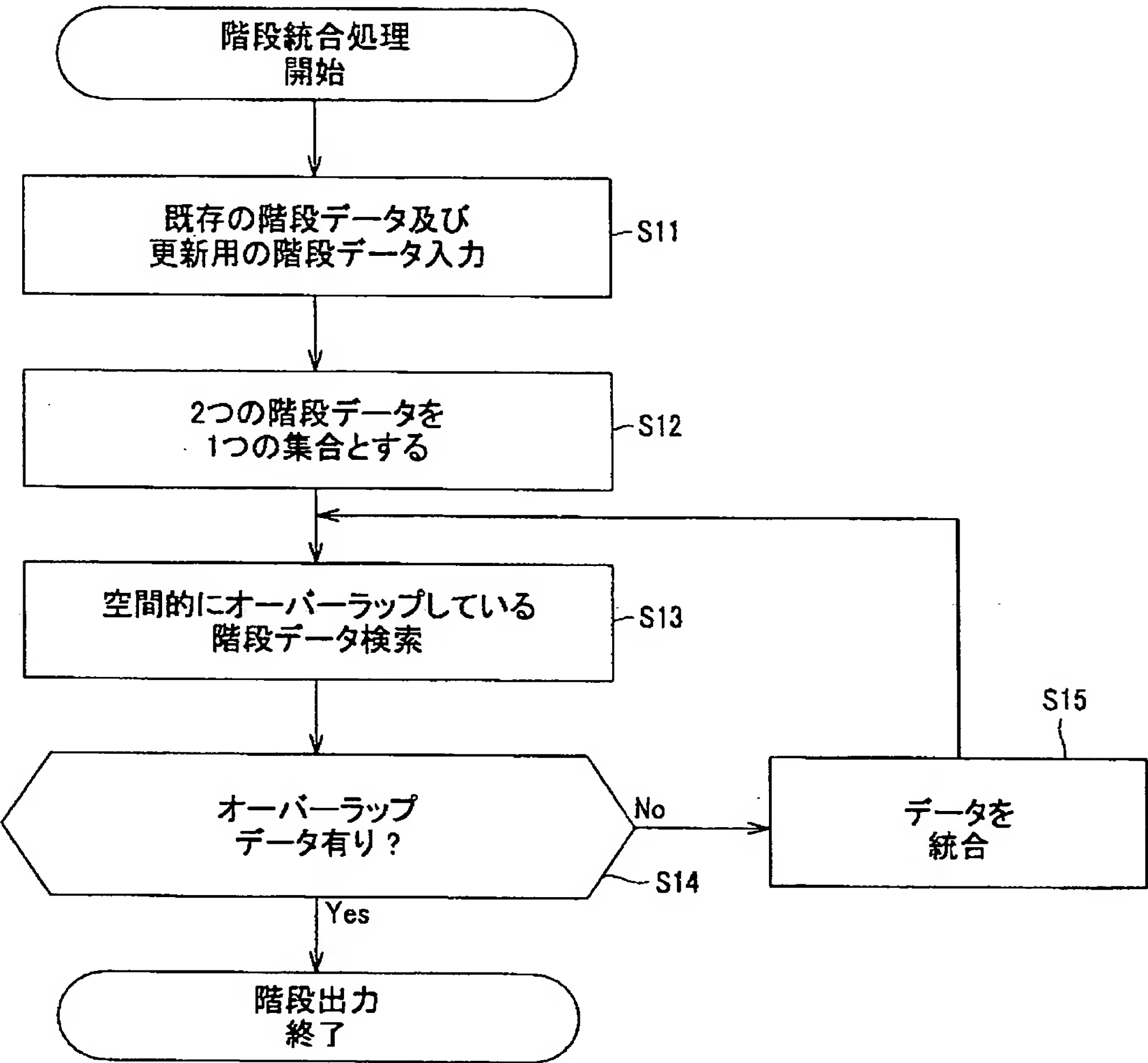


FIG.25



[図26]

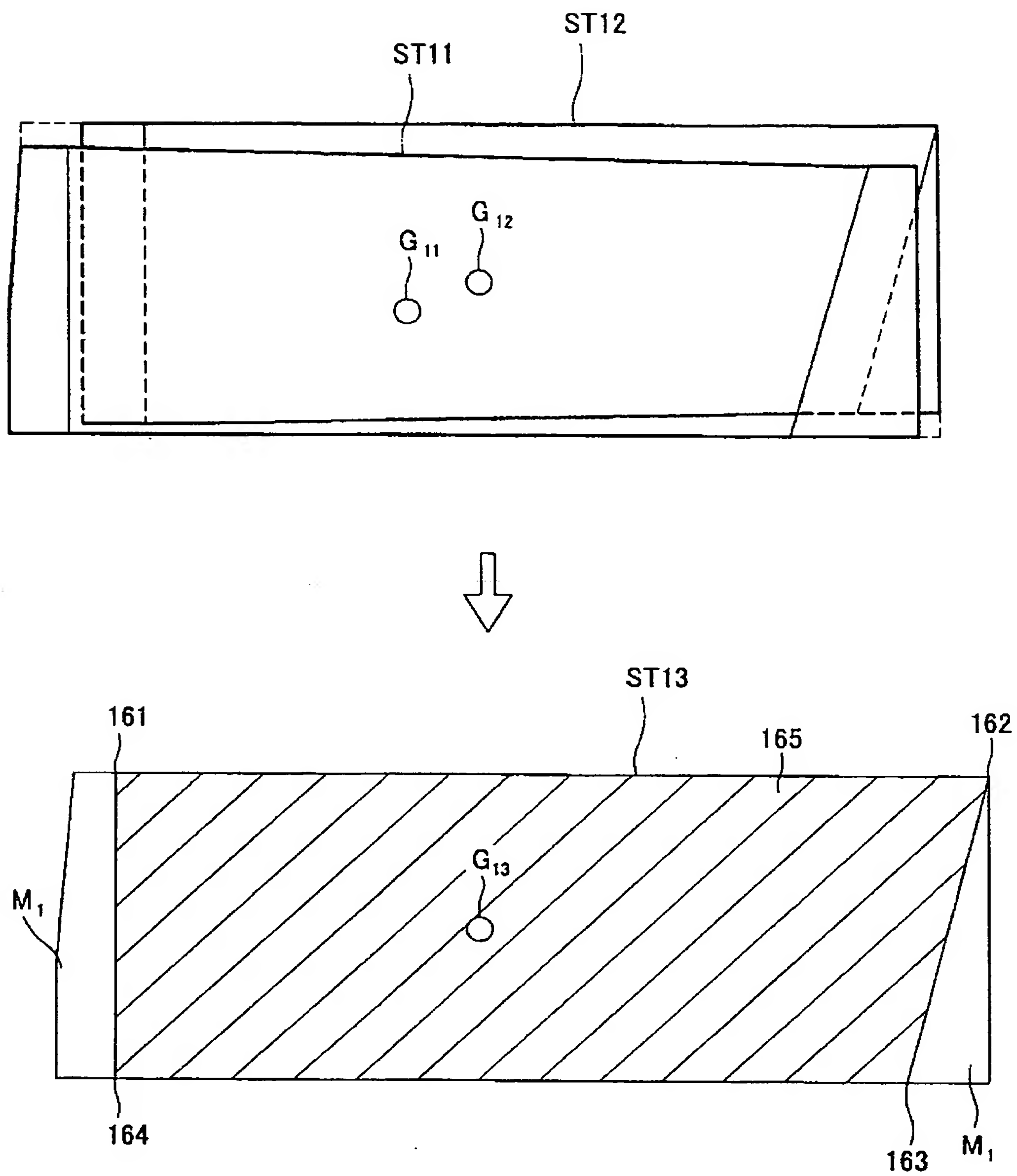

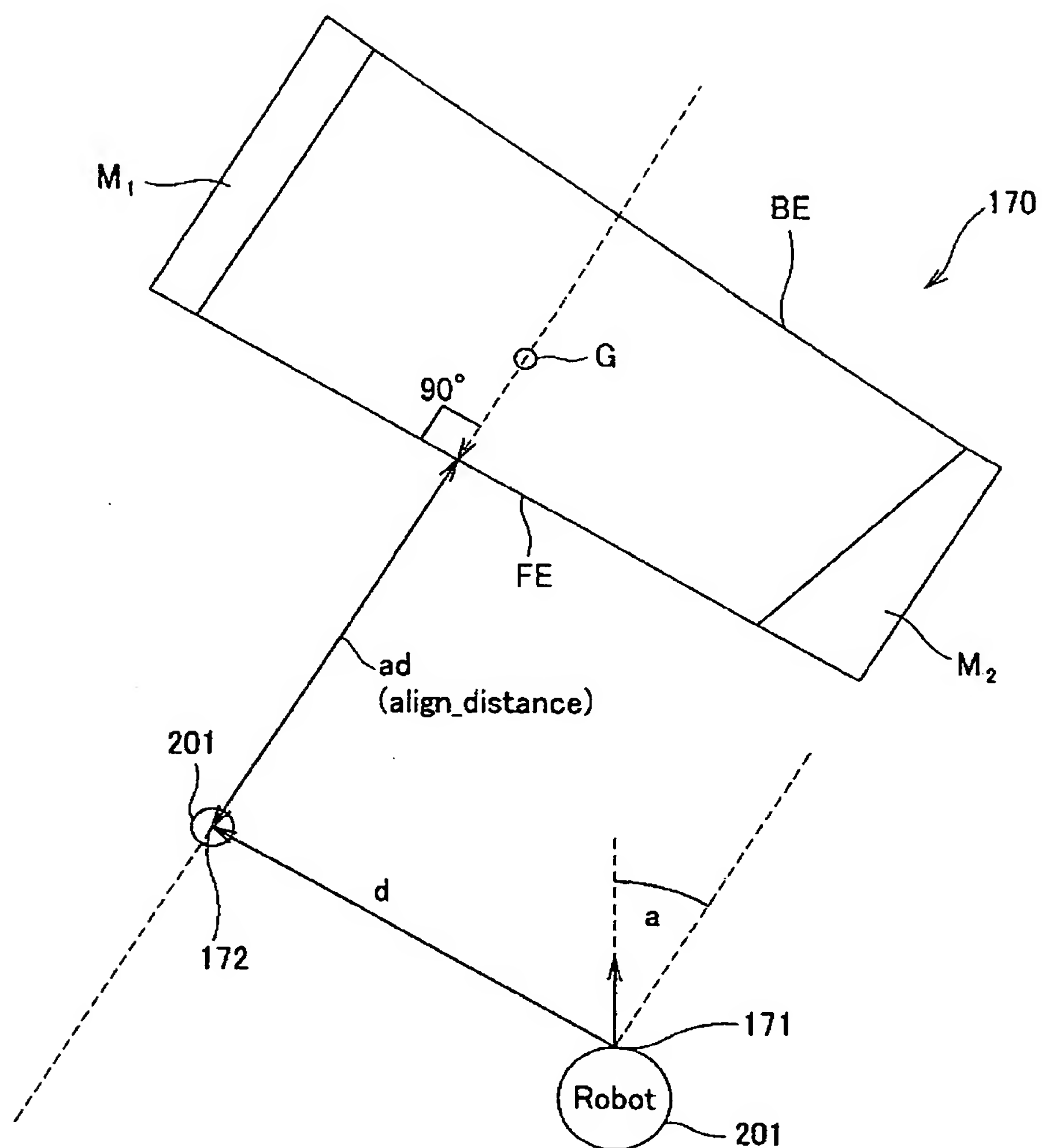


FIG.26

[ 27]



**FIG.27**

[図28]

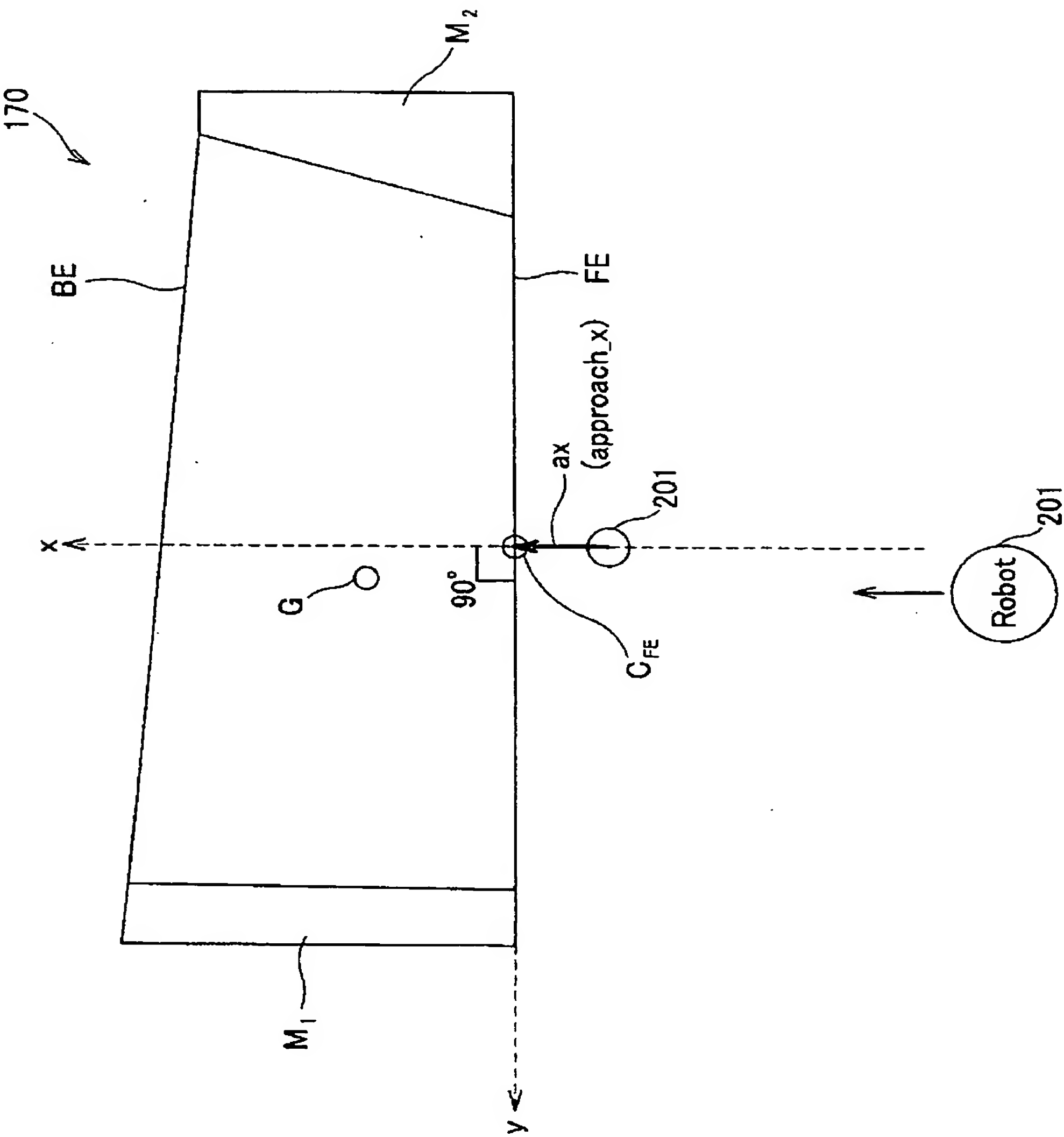


FIG.28

[図29]

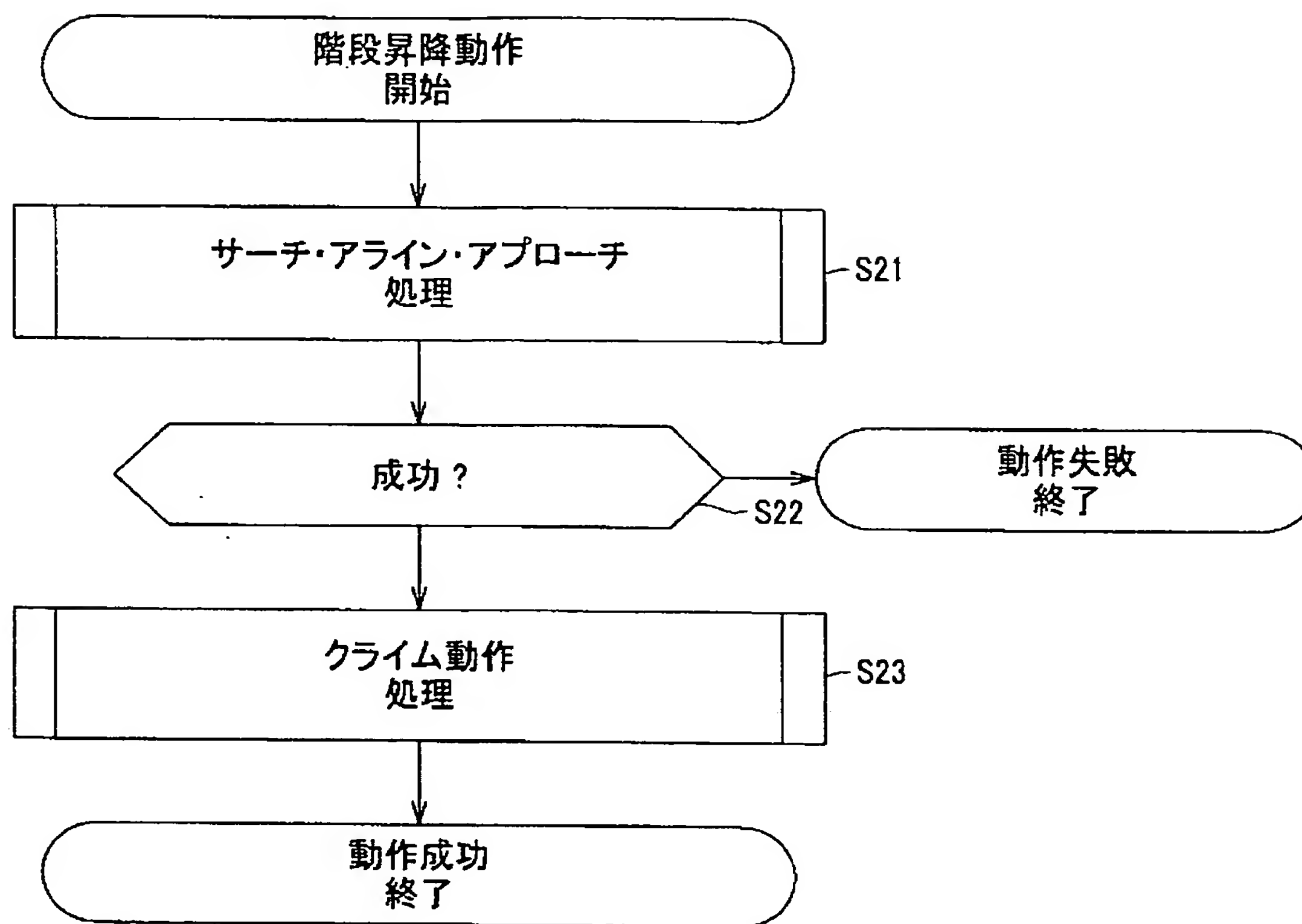


FIG.29

[図30]

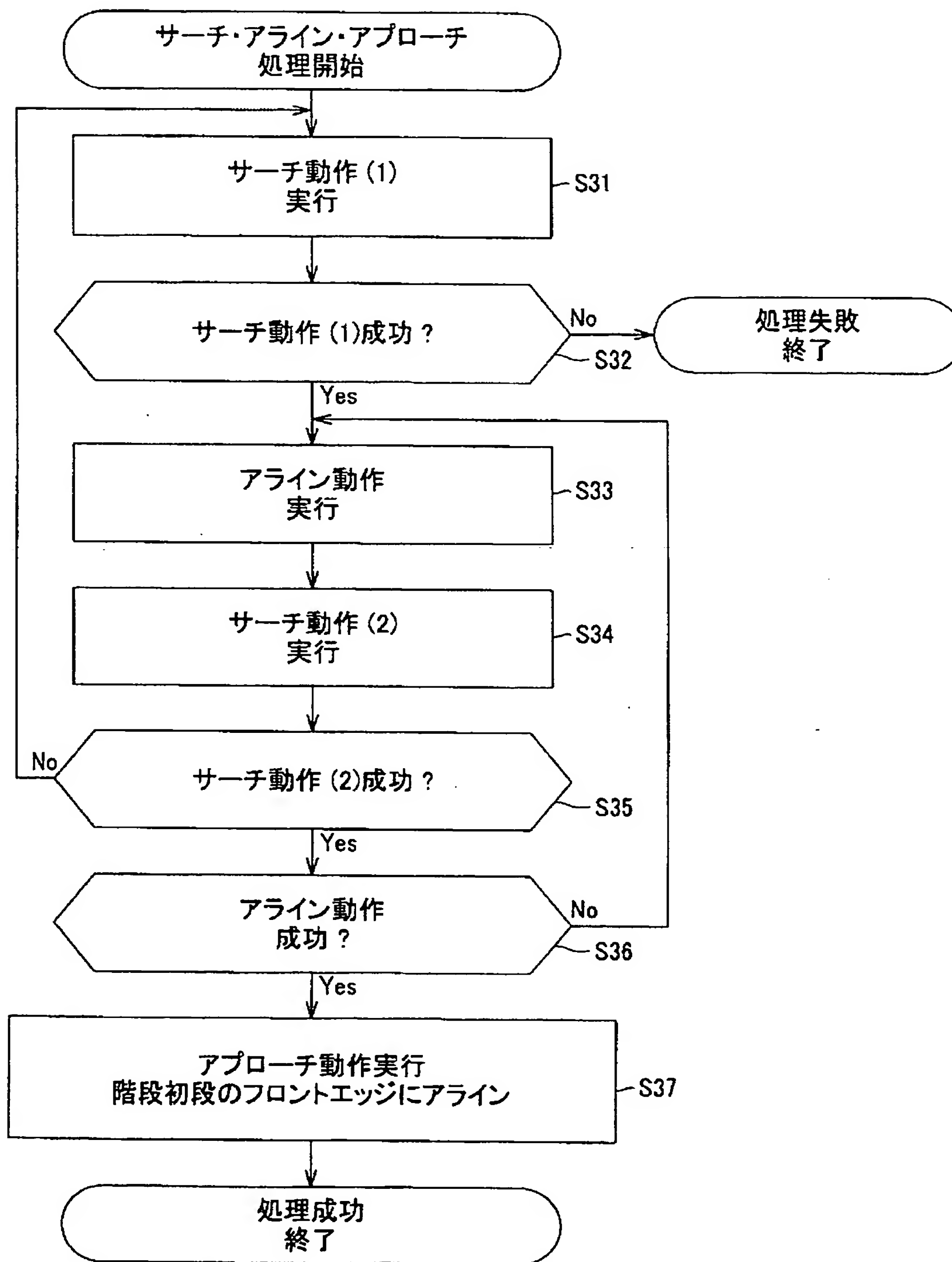


FIG.30

[図31]

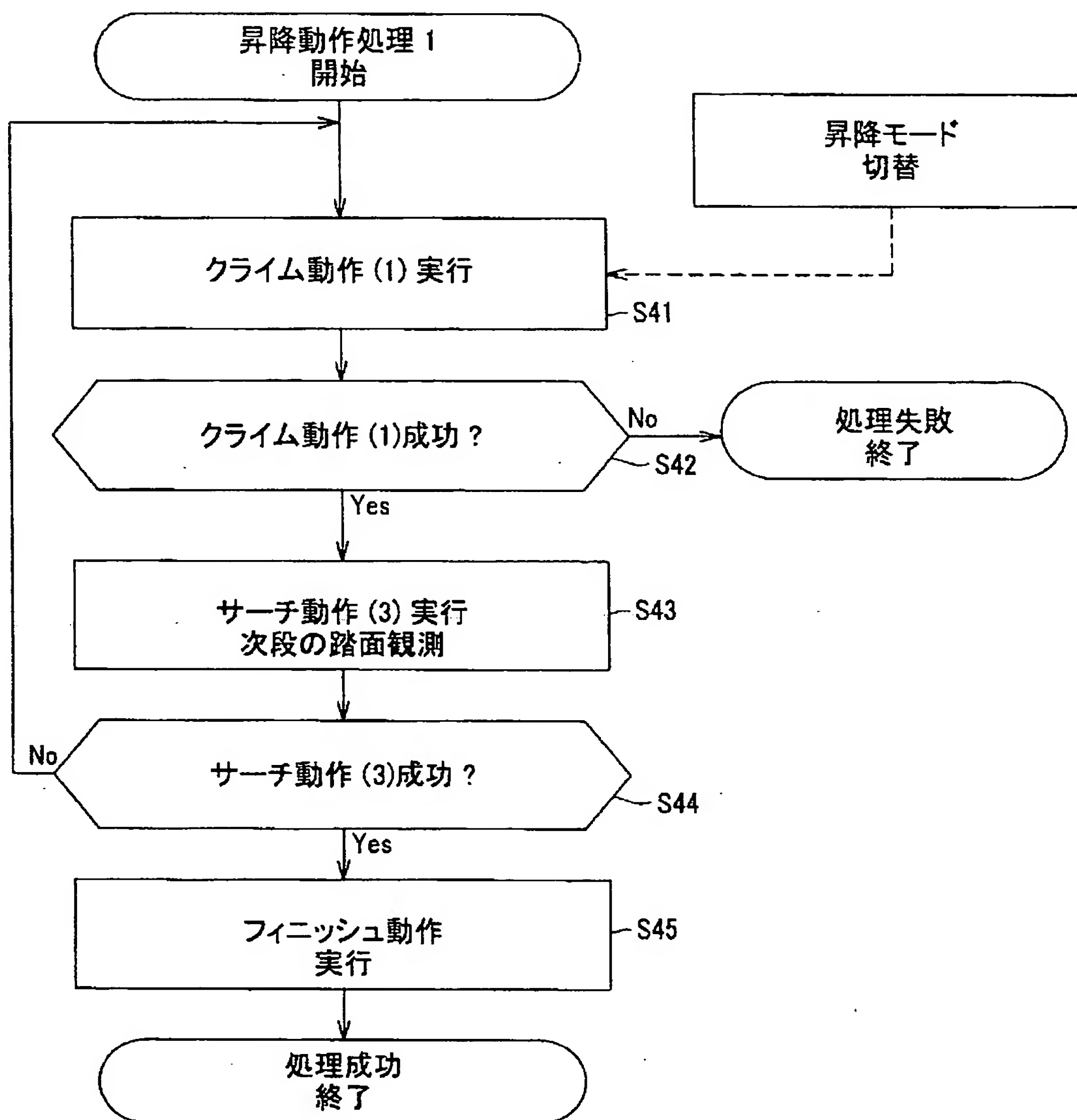


FIG.31



[図32]

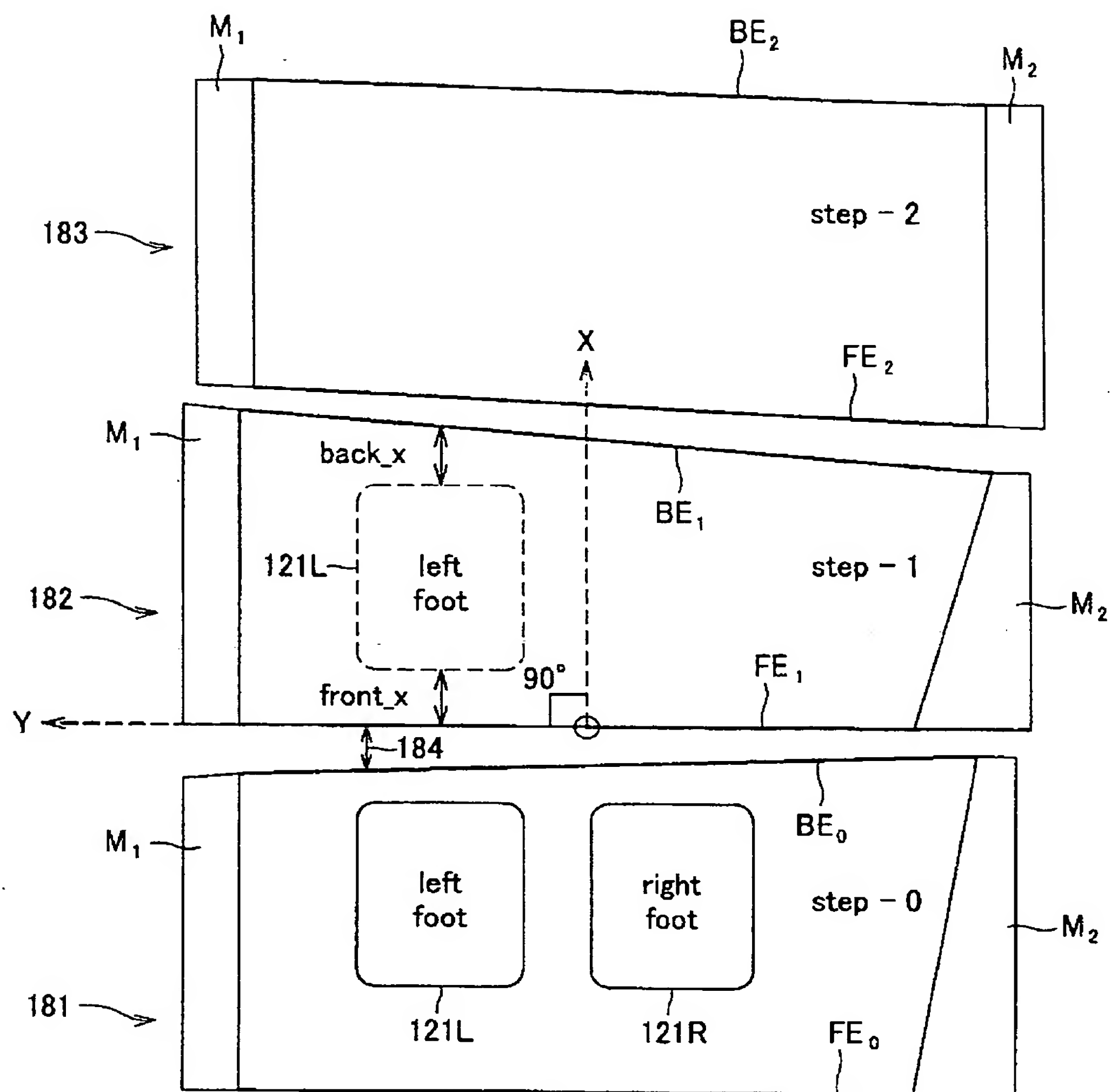


FIG.32

[図33]

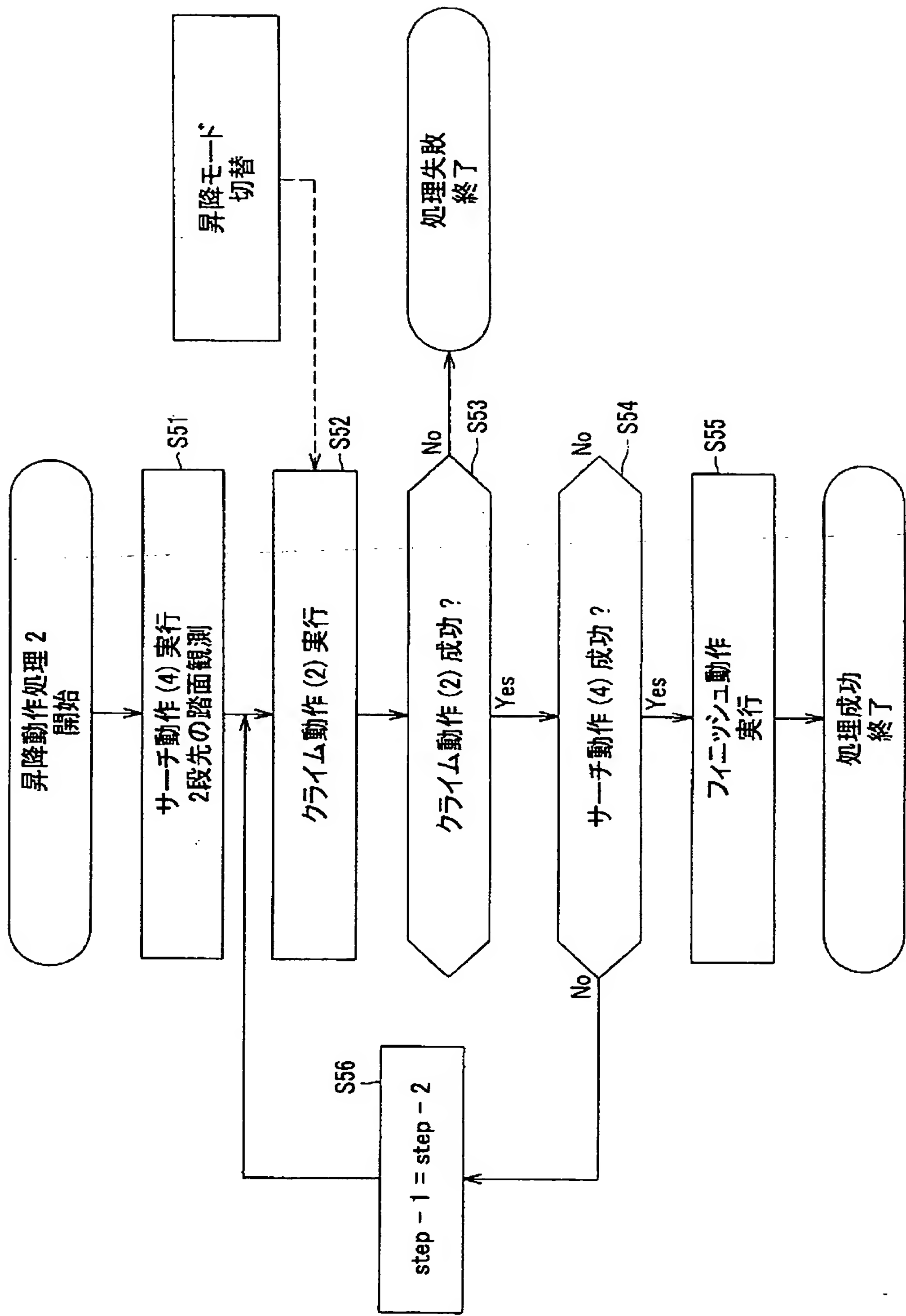


FIG.33

[図34]

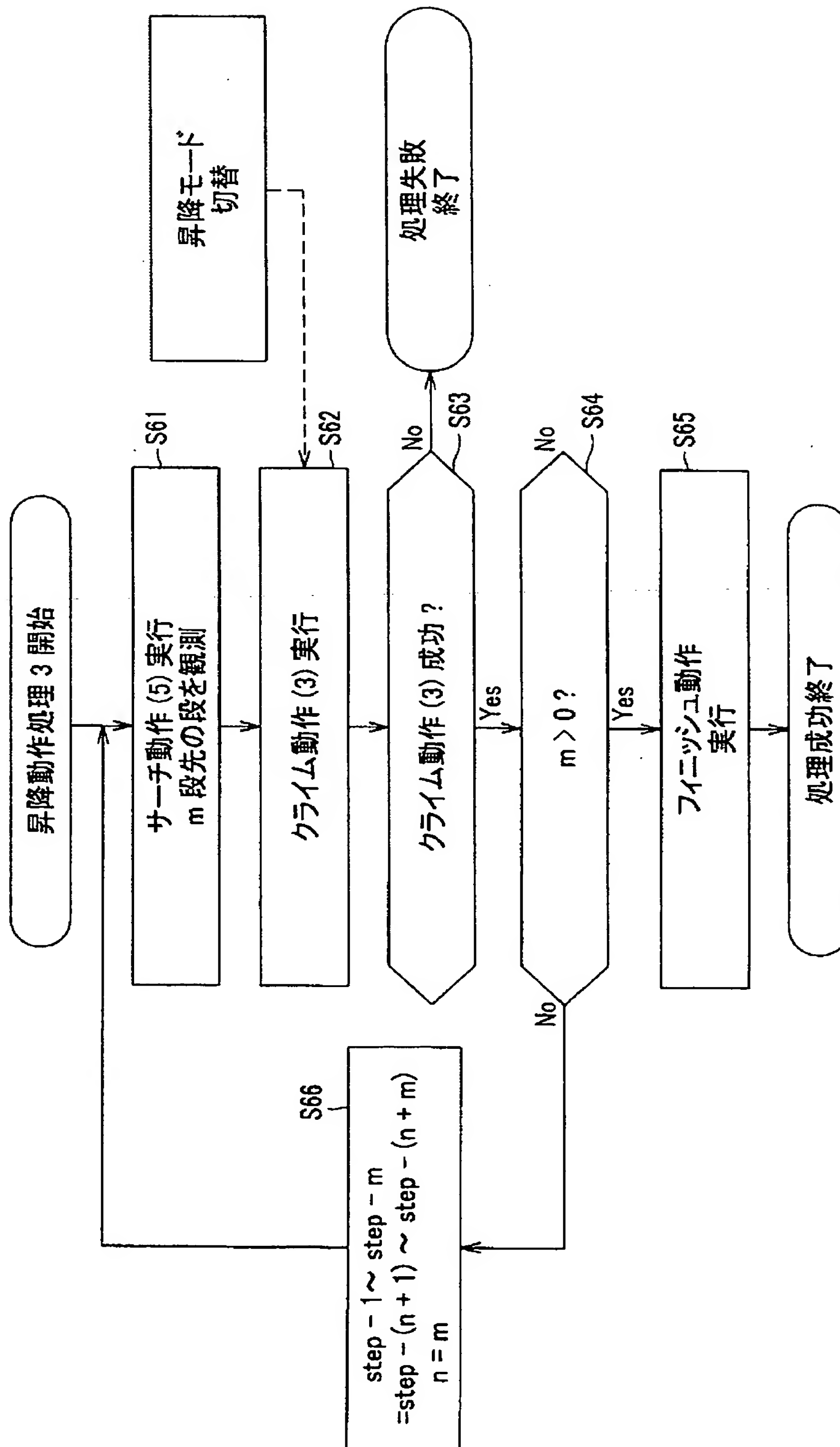
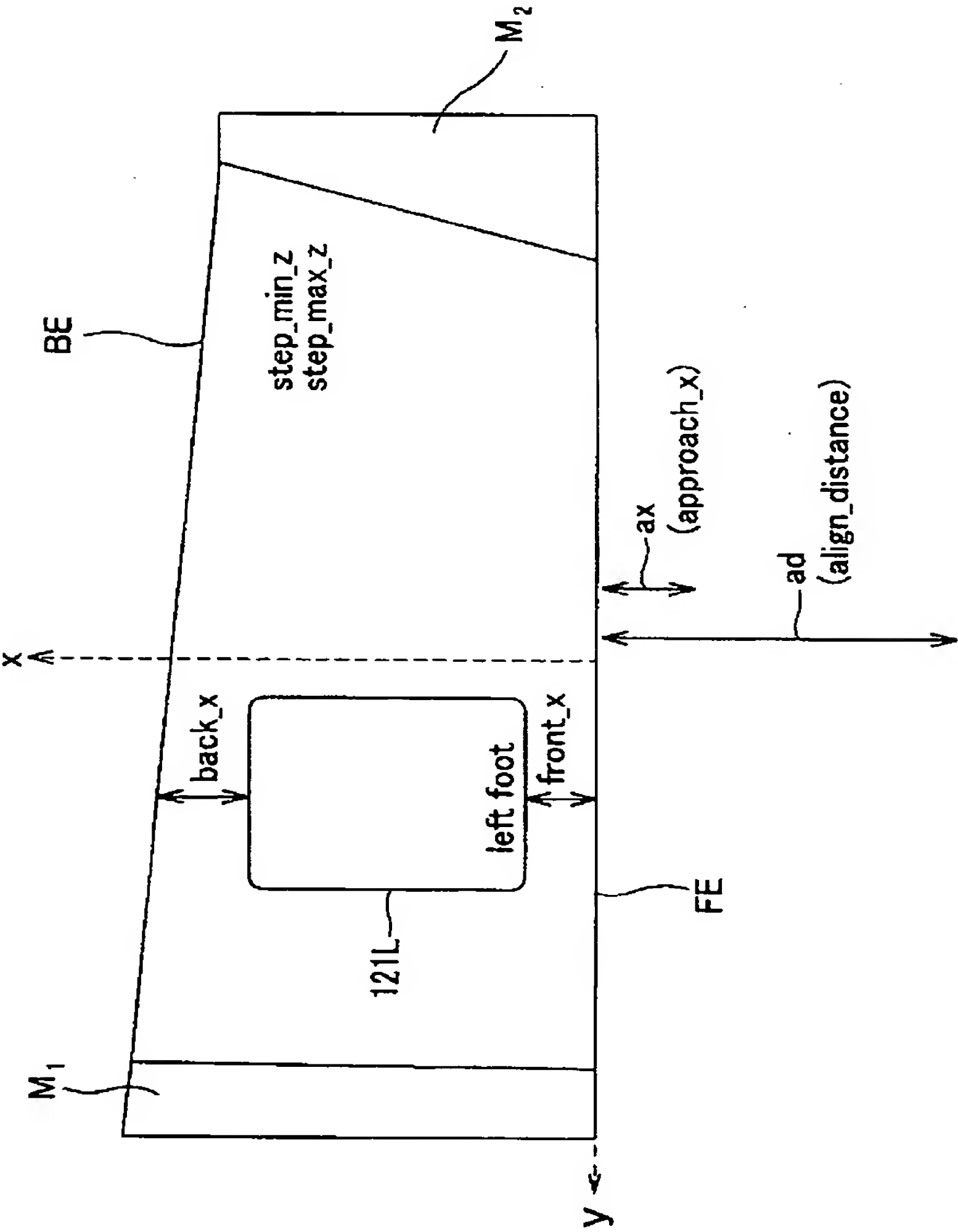


FIG.34

[図35]



	Ascend	Descend
step_min_z	15 mm	-50 mm
step_max_z	50 mm	-15 mm
align_distance	16 cm	5 cm
approach_x	-15 mm	5 mm
front_x_limit	-20 mm	5 mm
back_x_limit	-5 mm	10 mm
back_x_desired	-15 mm	5 mm

FIG.35A

FIG.35B

[図36]

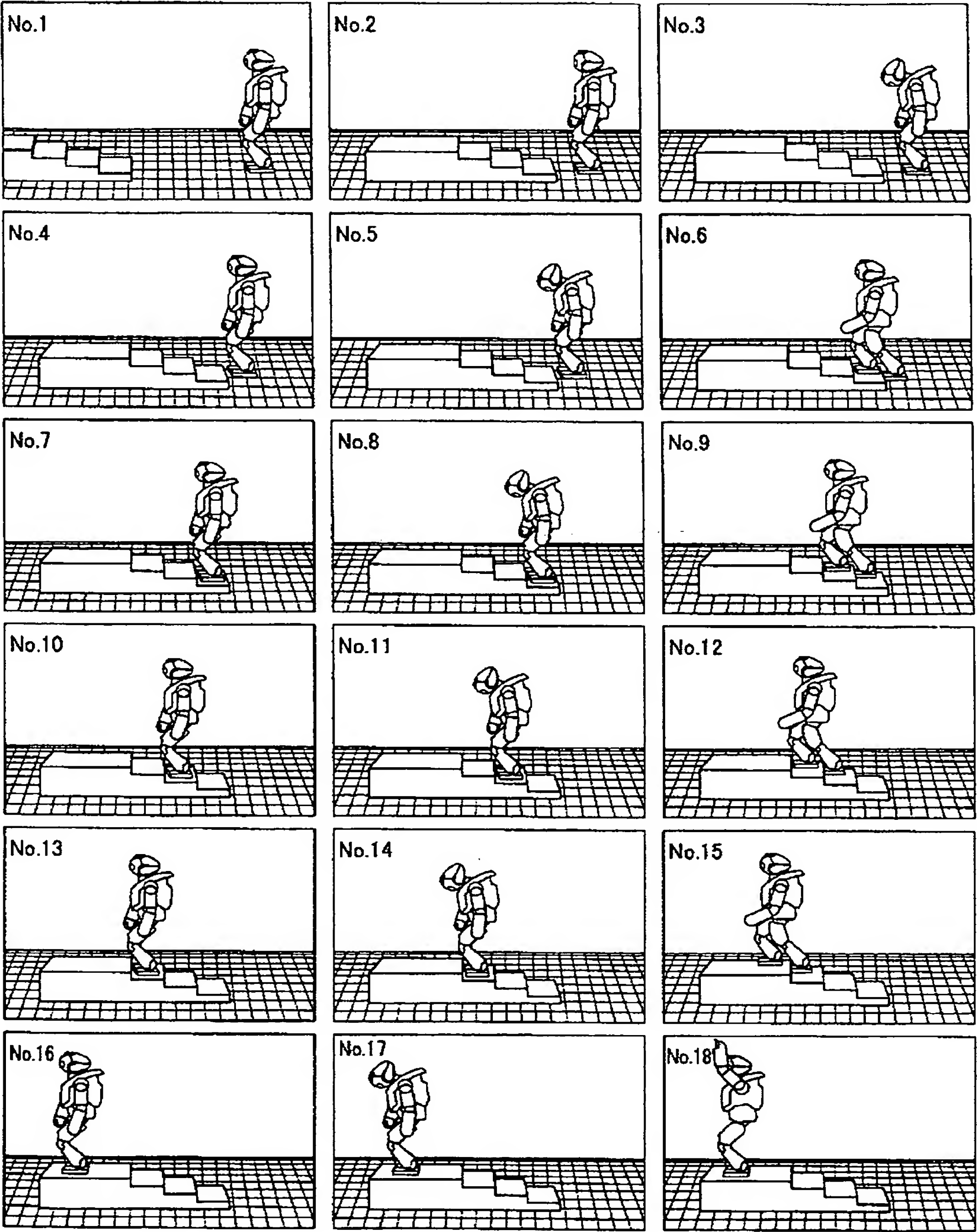


FIG.36

[図37]

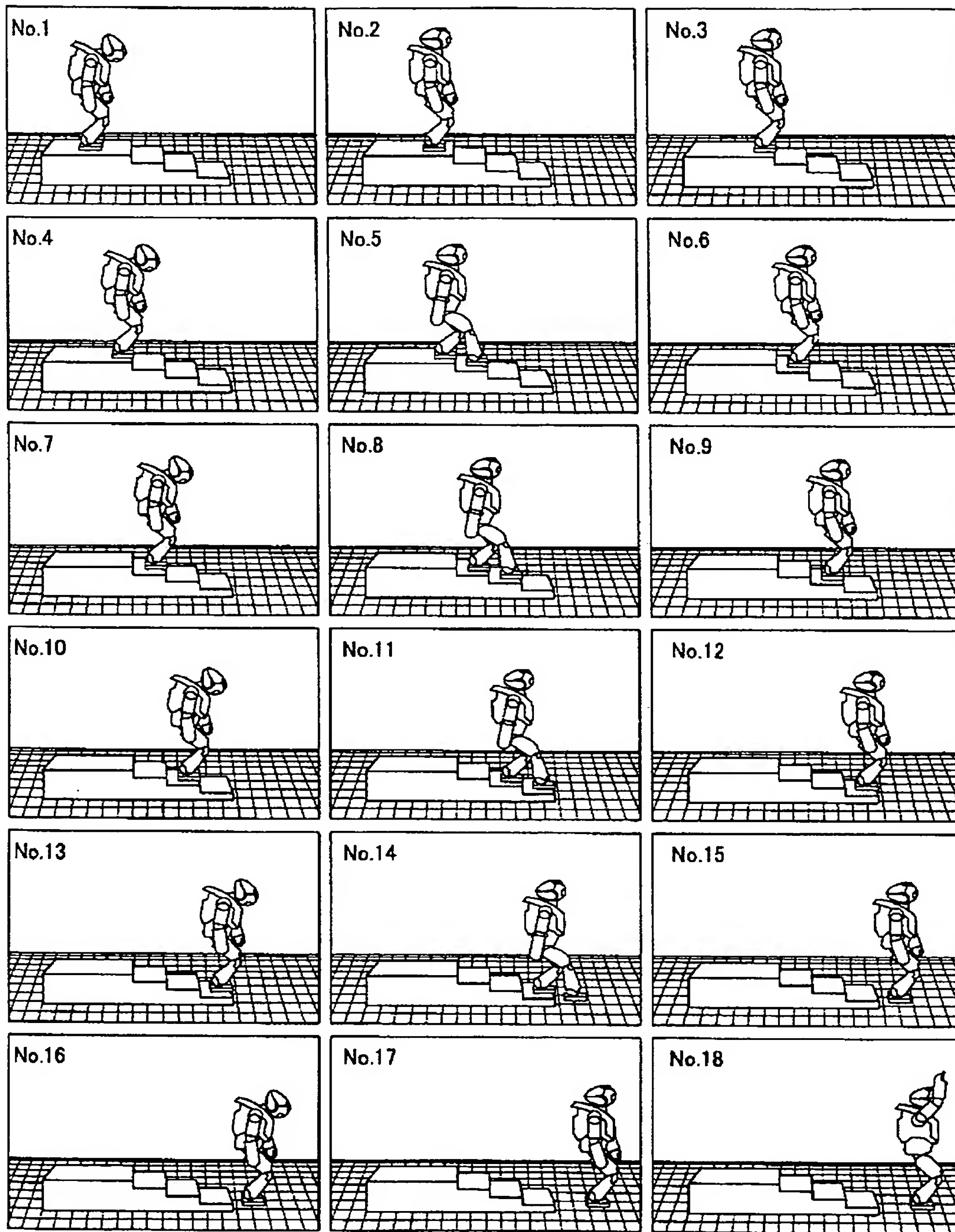


FIG.37



[図38]

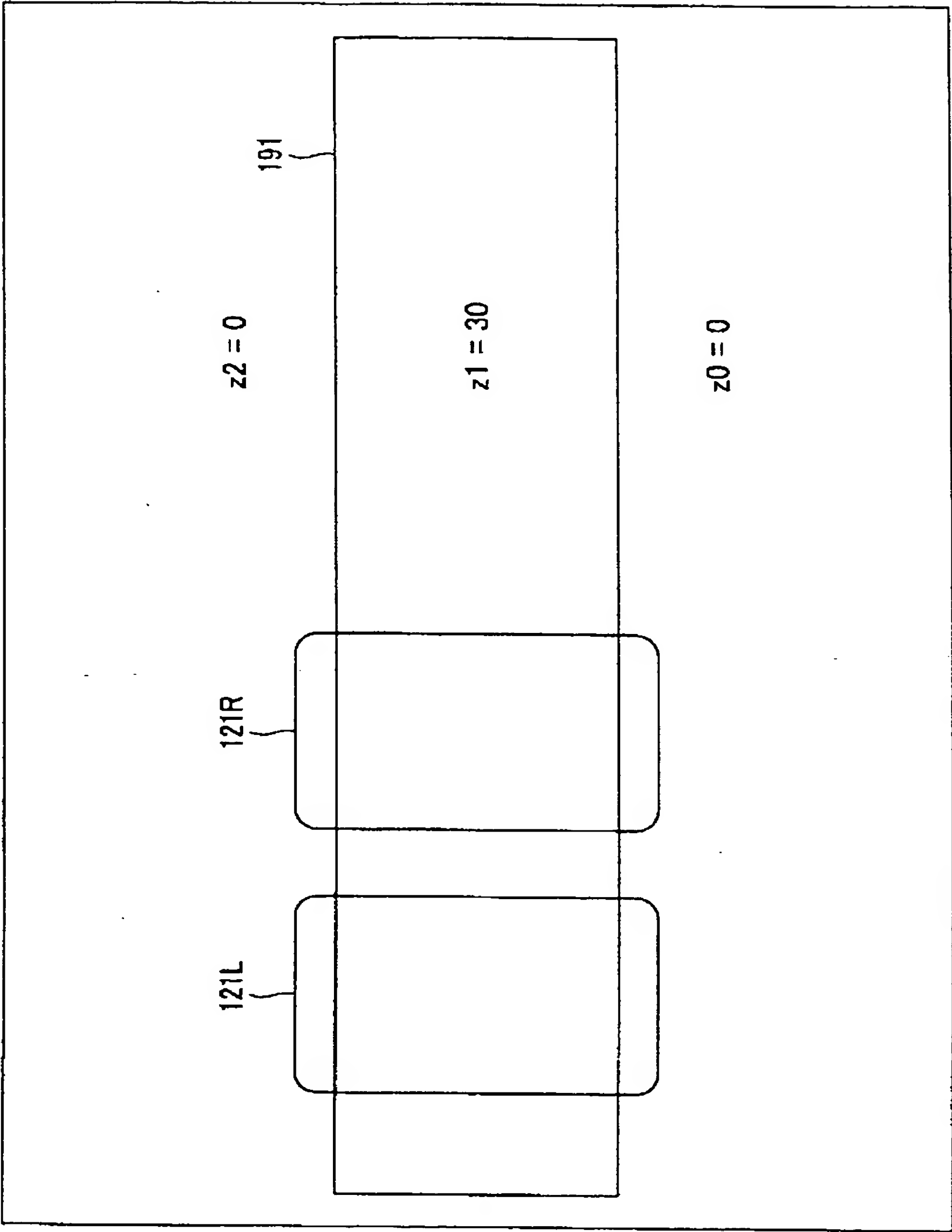


FIG.38

[図39]

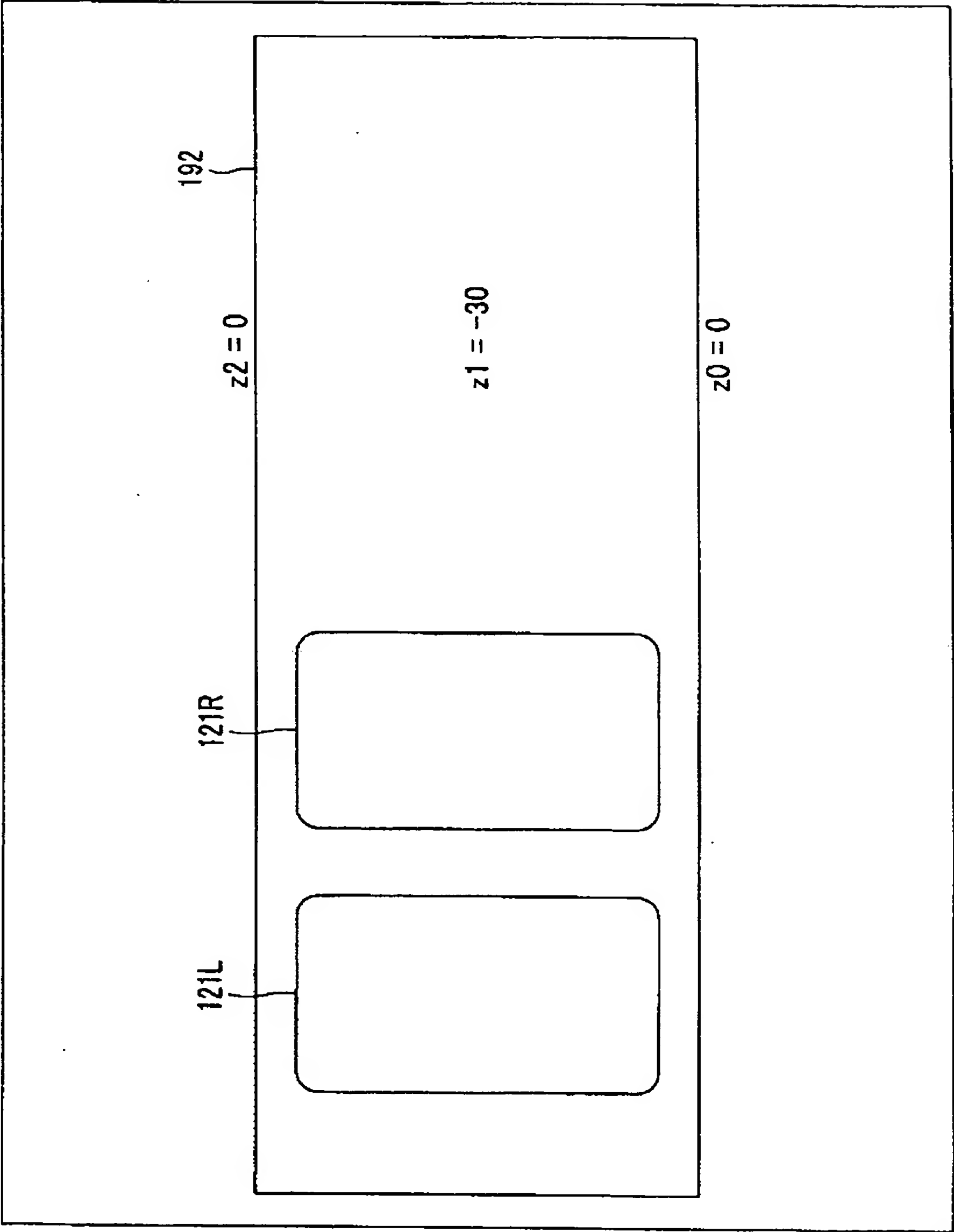


FIG.39

[図40]

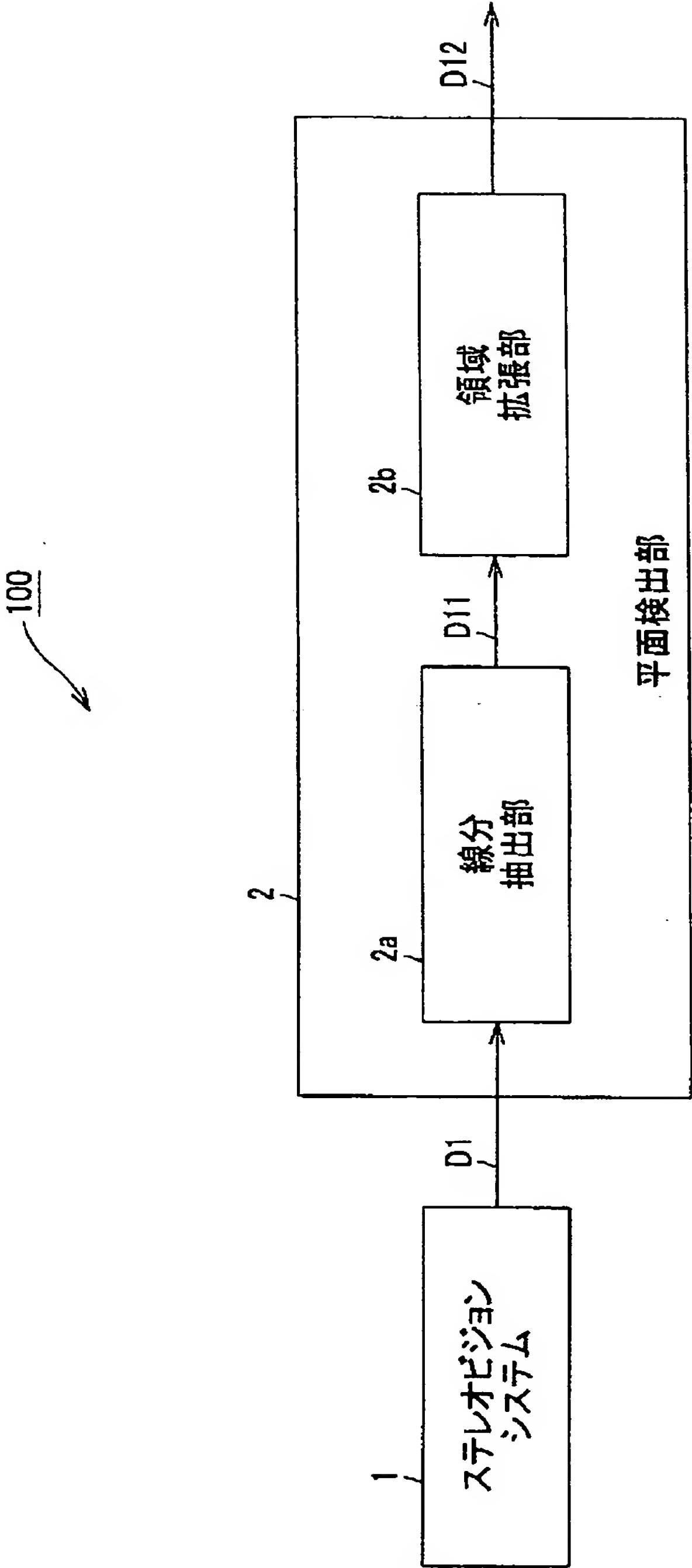


FIG.40

[図41]

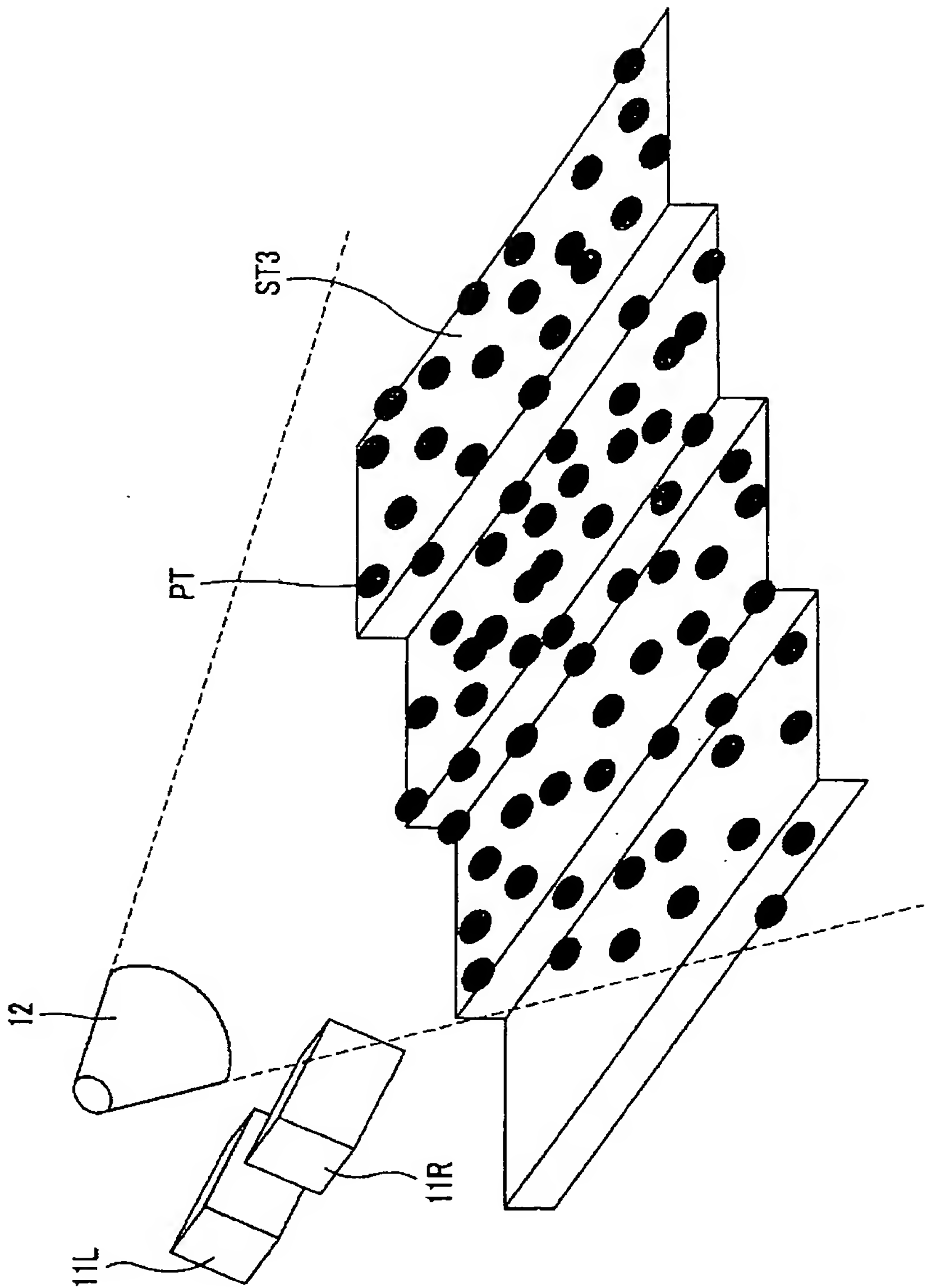


FIG.41

[図42]

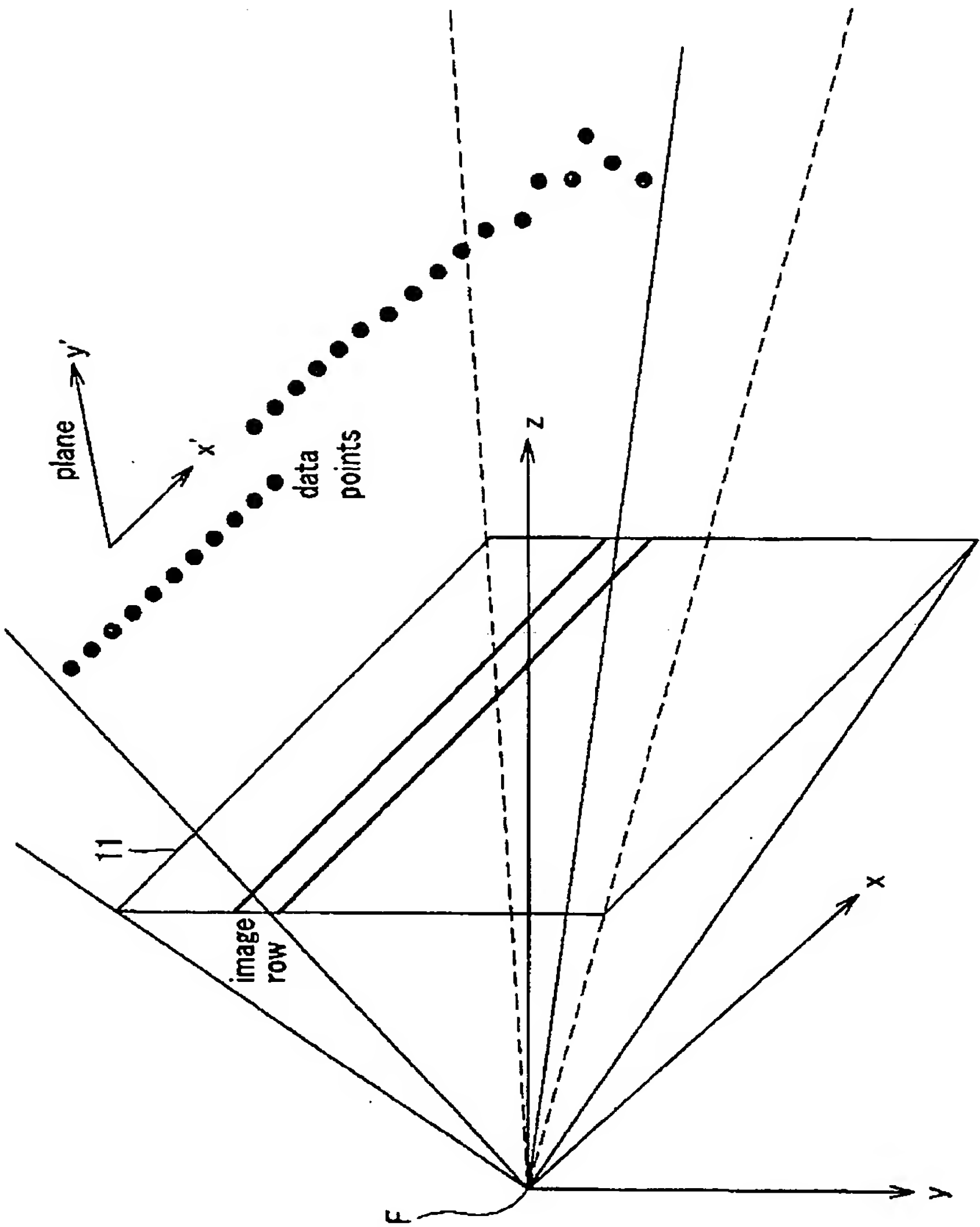


FIG.42

[図43]

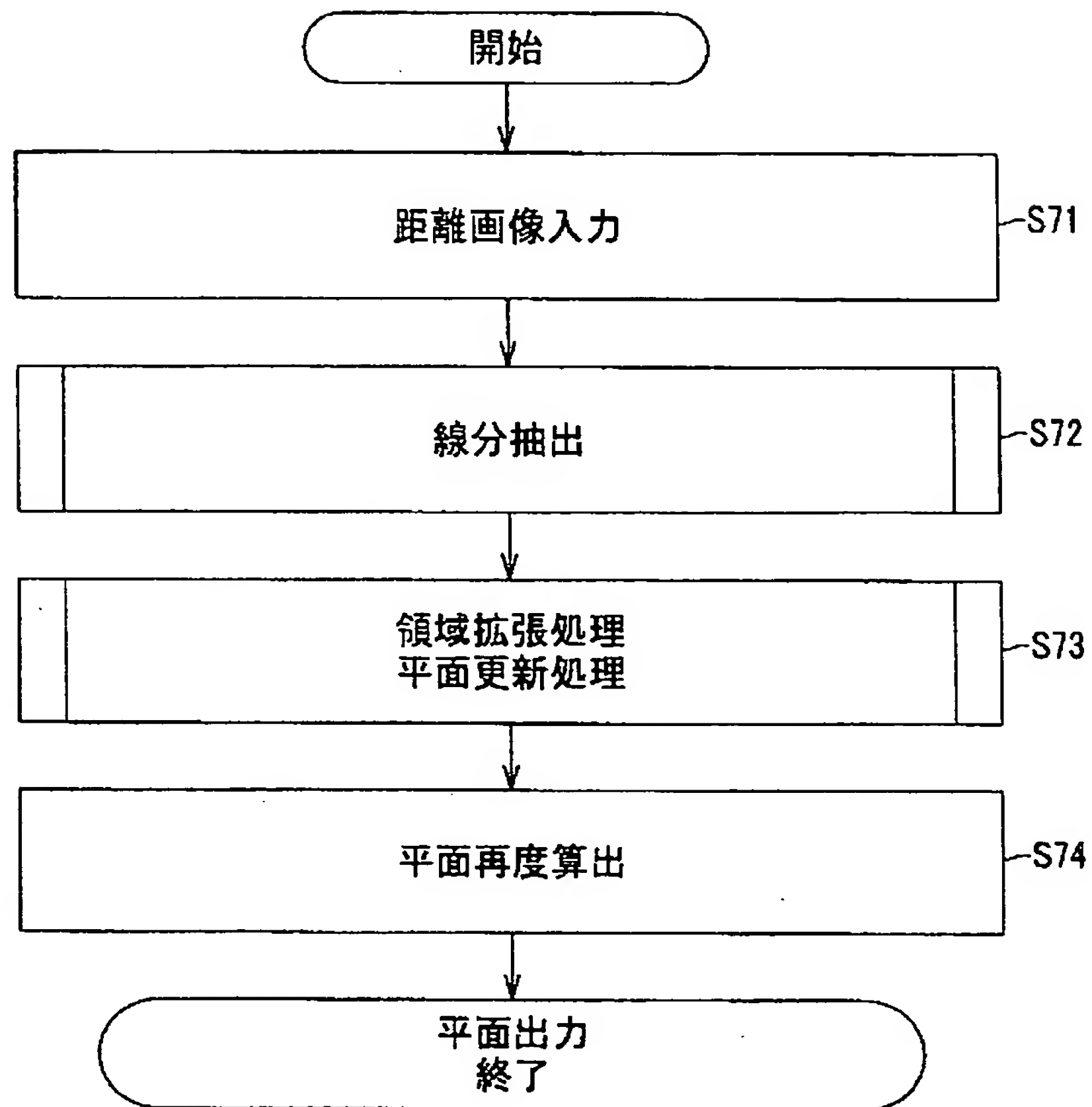


FIG.43



[図44]

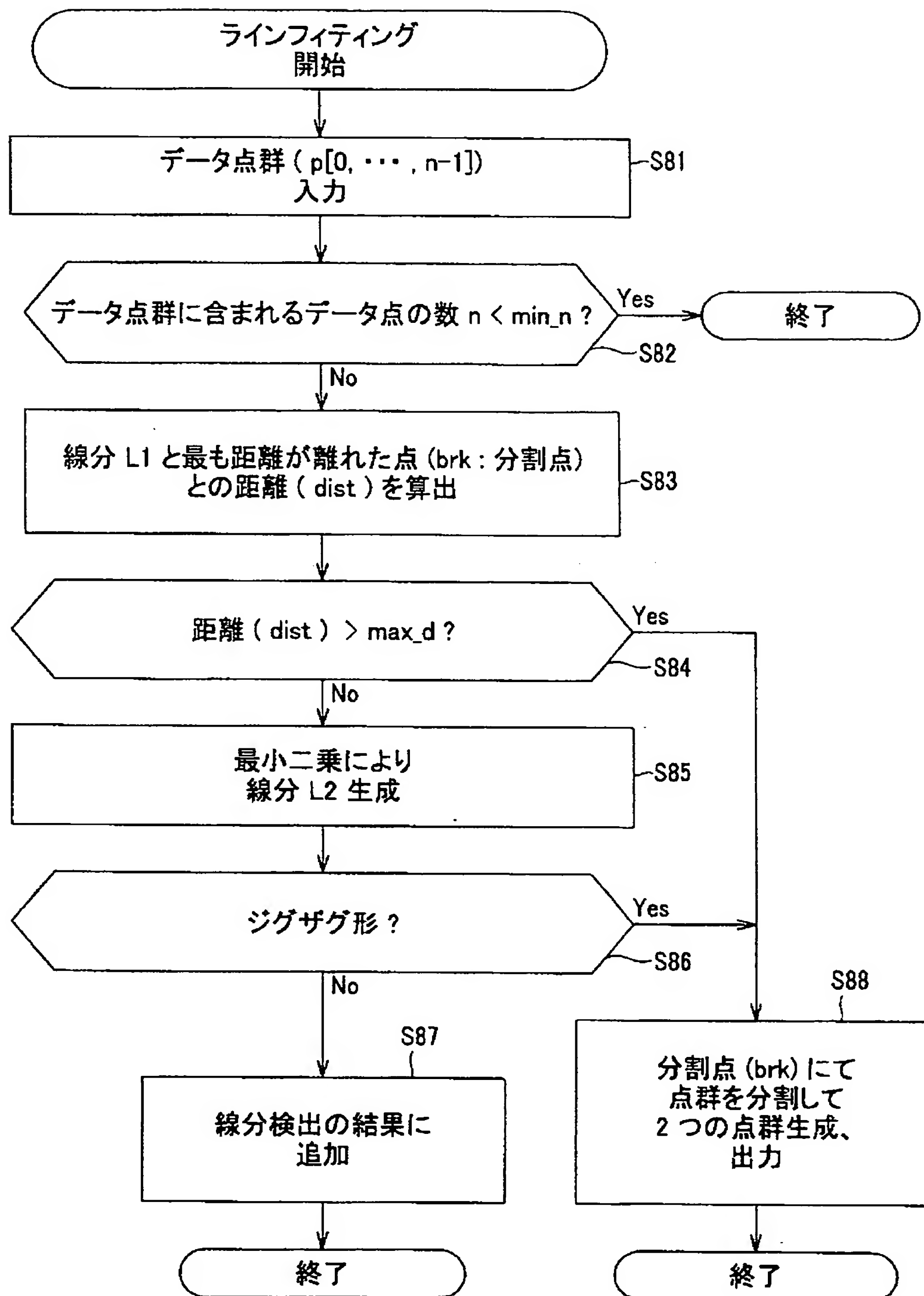


FIG.44

[図45]

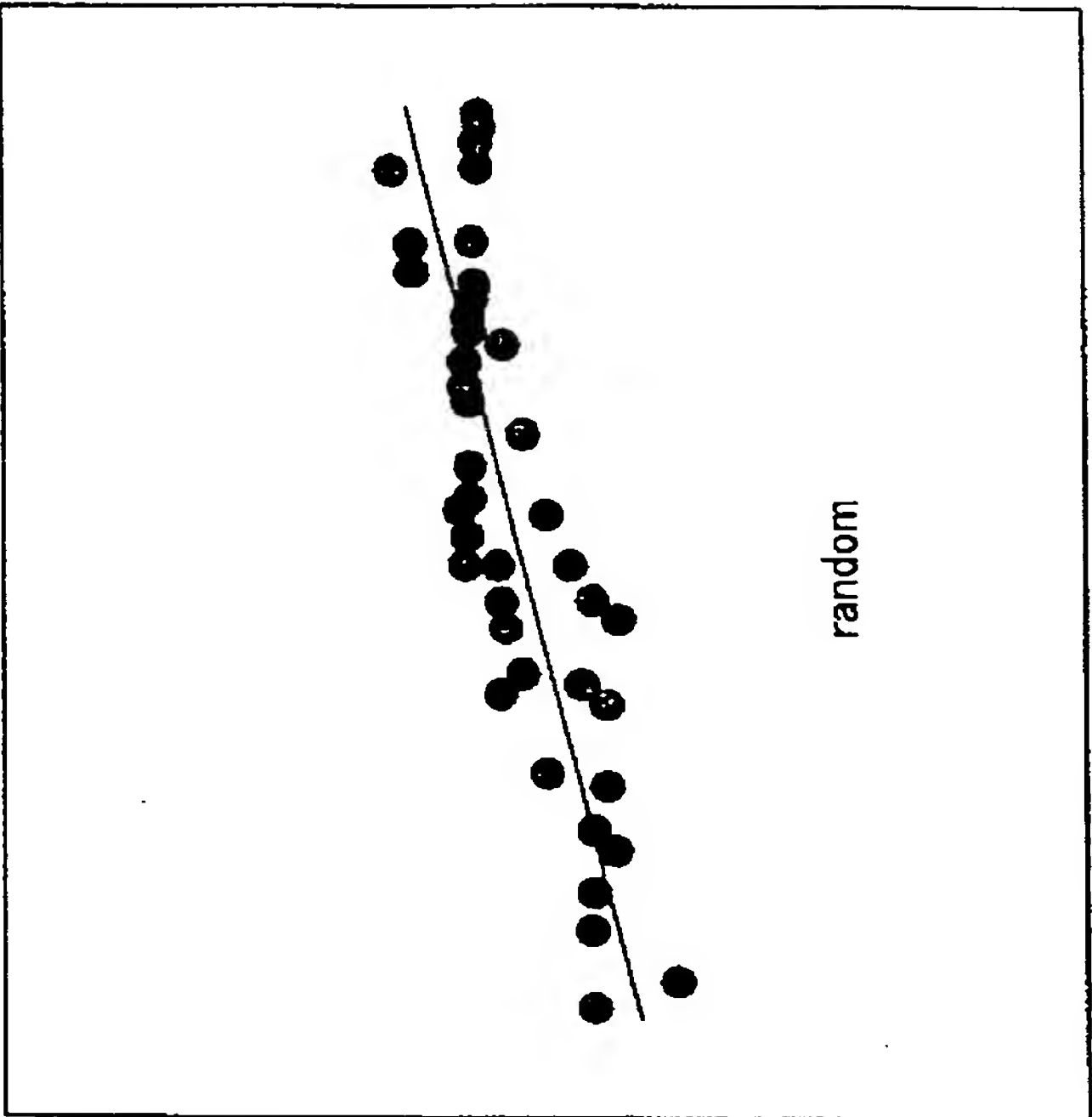


FIG.45B

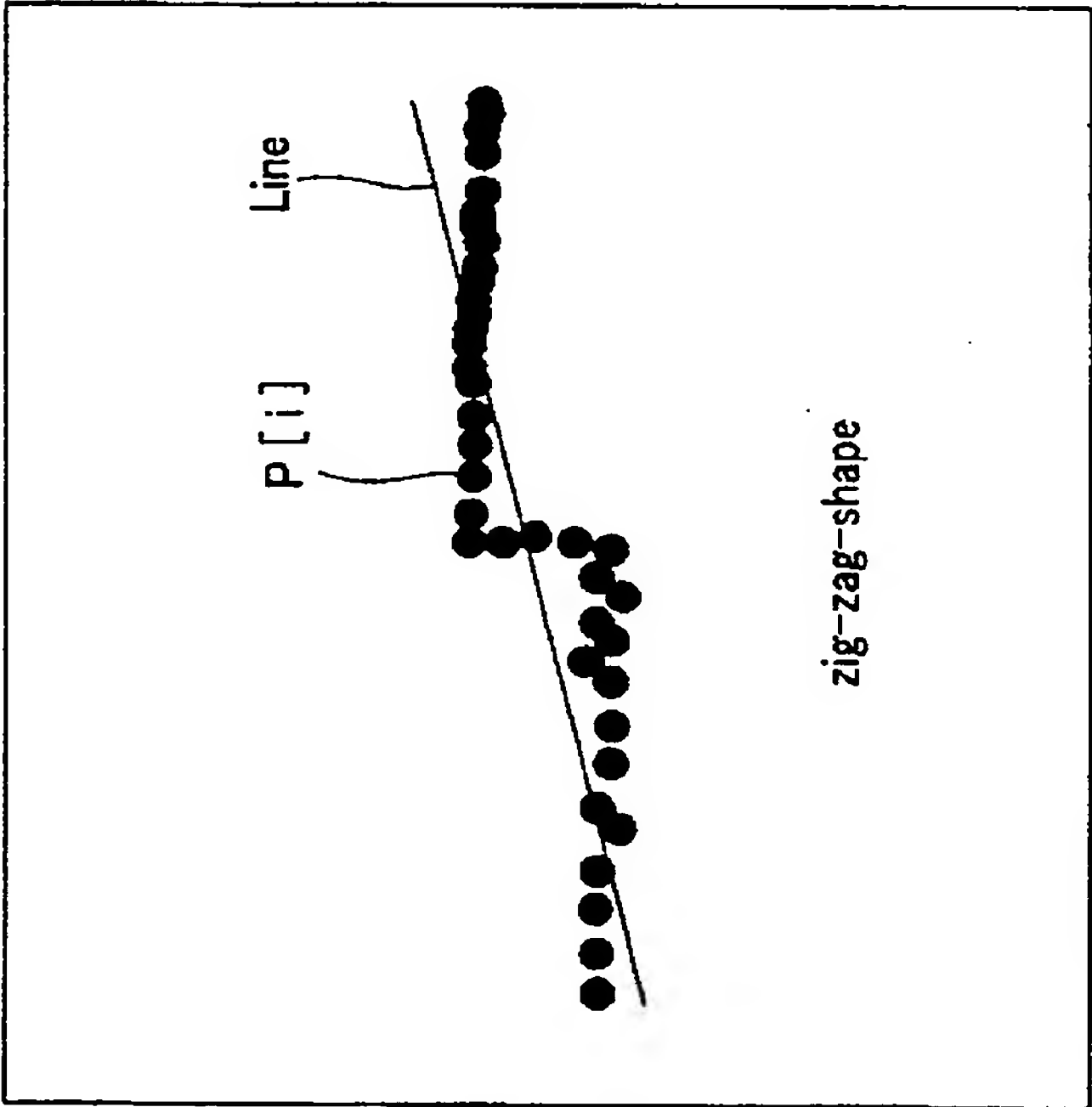


FIG.45A

[図46]

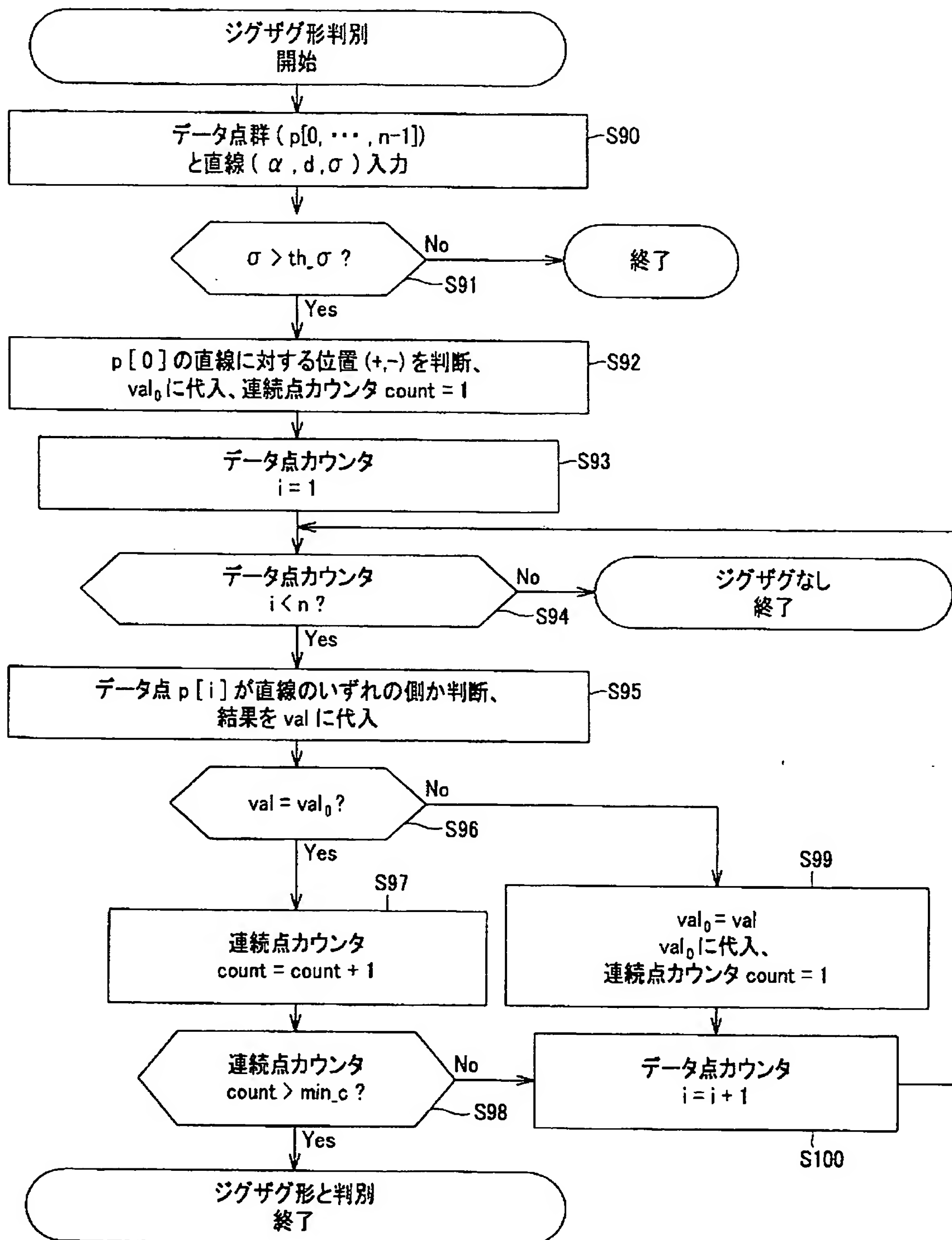


FIG.46

[図47]

Input:  $pts$  : vector of points,  $n$  : number of points,  $\alpha, d, \sigma$  : parameters and std-dev of fitted line.

Output : true if curve contains a *zig-zag*- shape, false otherwise.

Sequence :

```
    if  $\sigma > thresh\sigma$  then
       $val0 = pts[0].x * \cos \alpha + pts[0].y * \sin \alpha + d$ 
       $count = 1$ 
      for  $i = 1$  to  $(n-1)$  do
         $val = pts[i].x * \cos \alpha + pts[i].y * \sin \alpha + d$ 
        if ( $val * val0 \leq 0$ ) then
           $val0 = val$ 
           $count = 1$ 
        else
           $count = count + 1$ 
          if ( $count \geq min-points-for-zig-zag-shape$ ) then
            return true
          endif
        endif
      endfor
    endif
    return false
```

**FIG.47**

[図48]

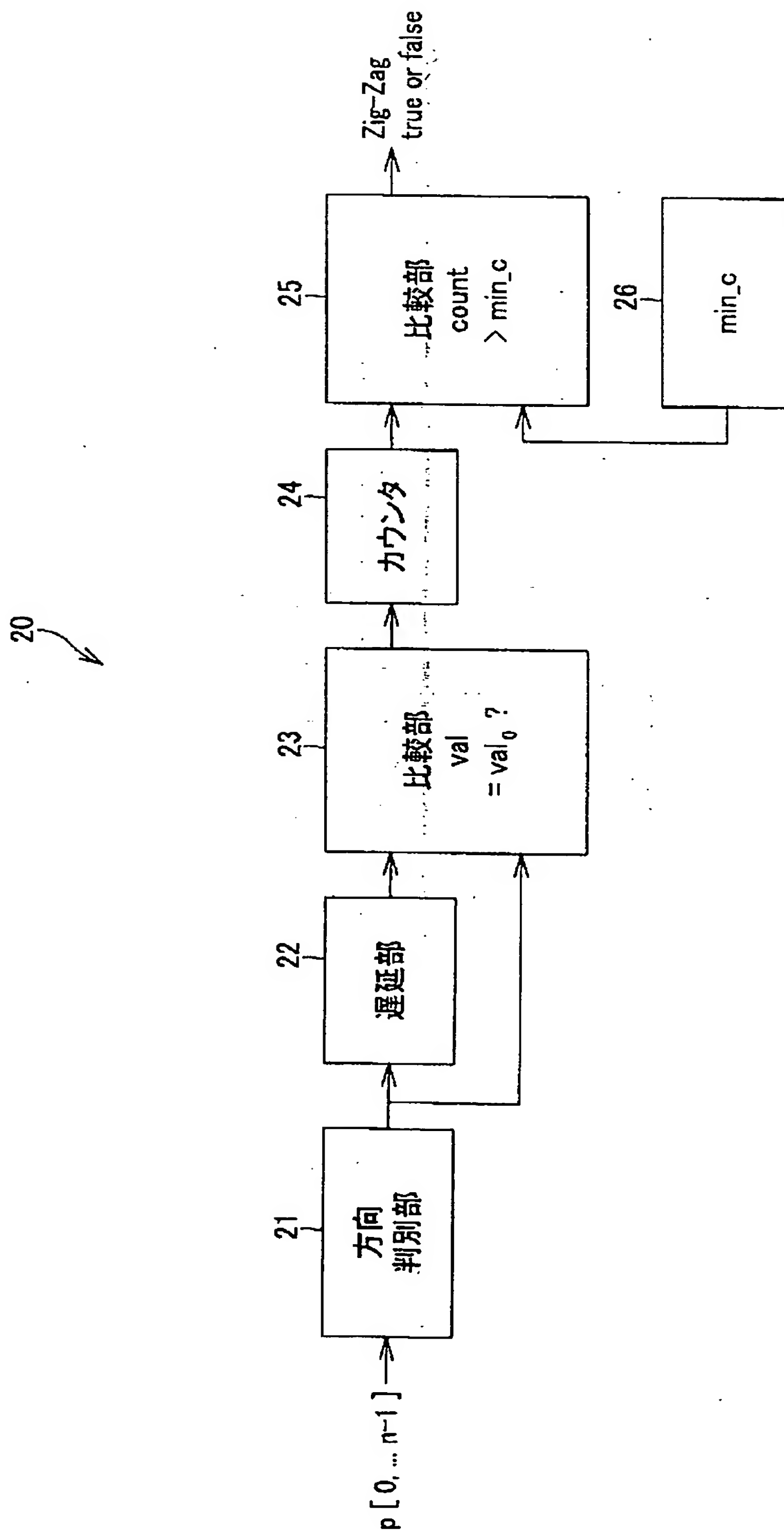


FIG.48

[図49]

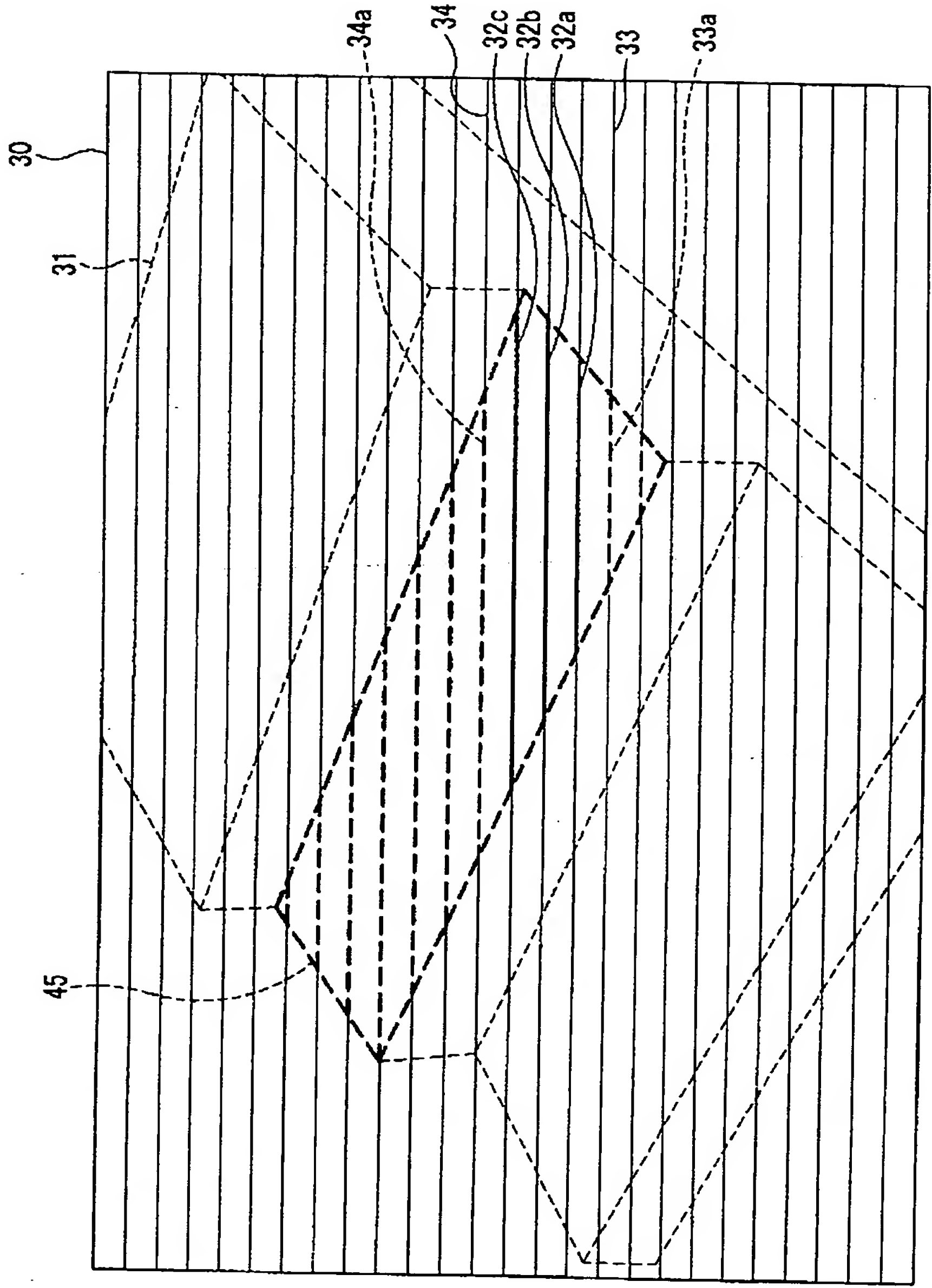


FIG. 49



[図50]

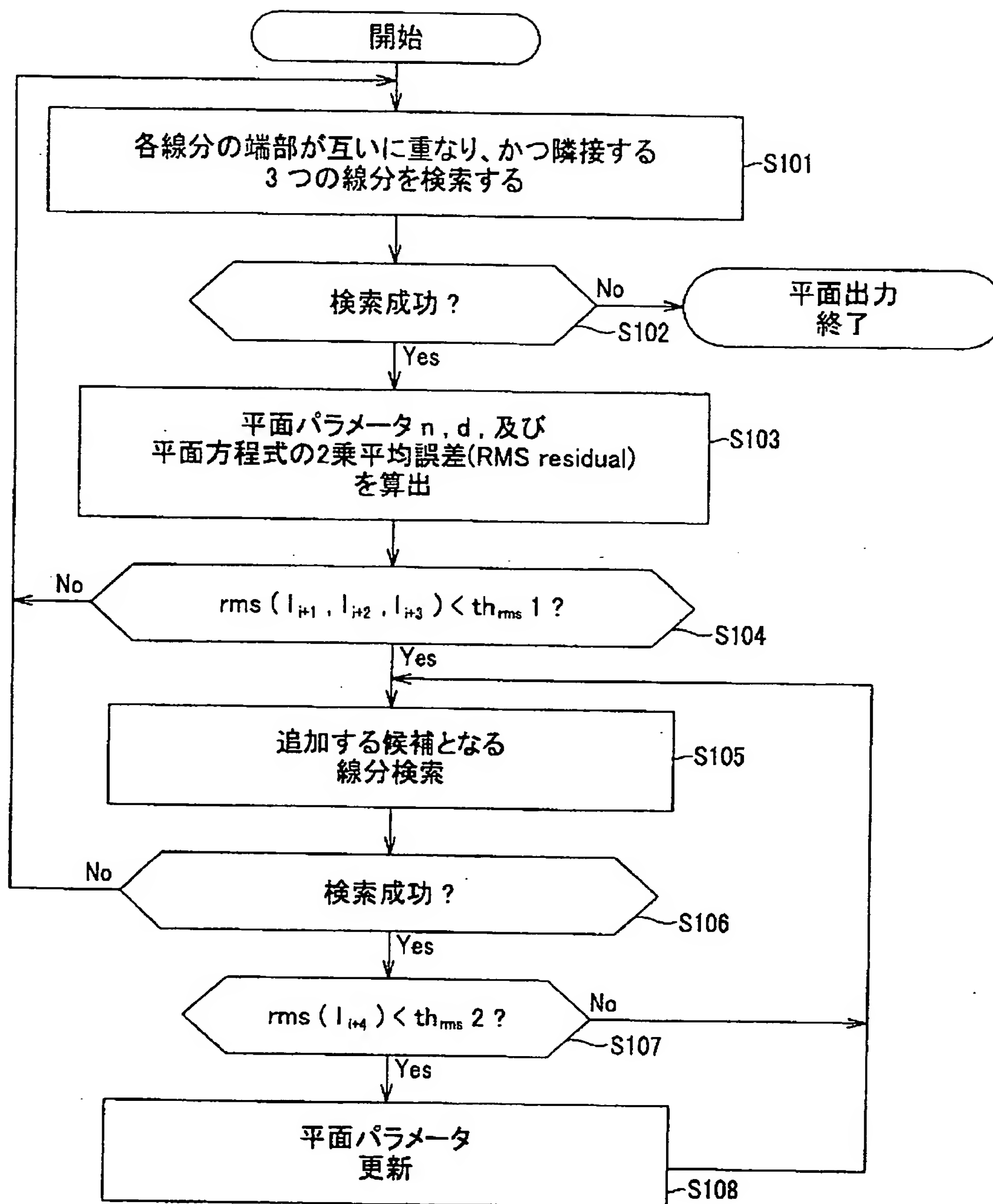


FIG.50

[図51]

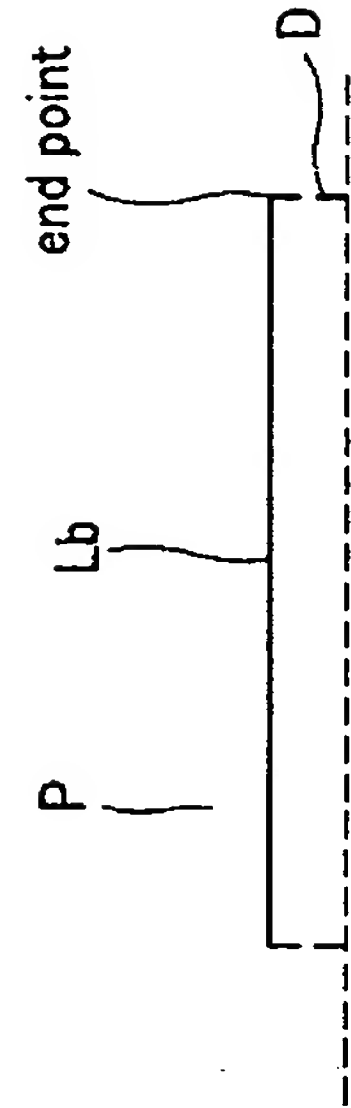


FIG.51B

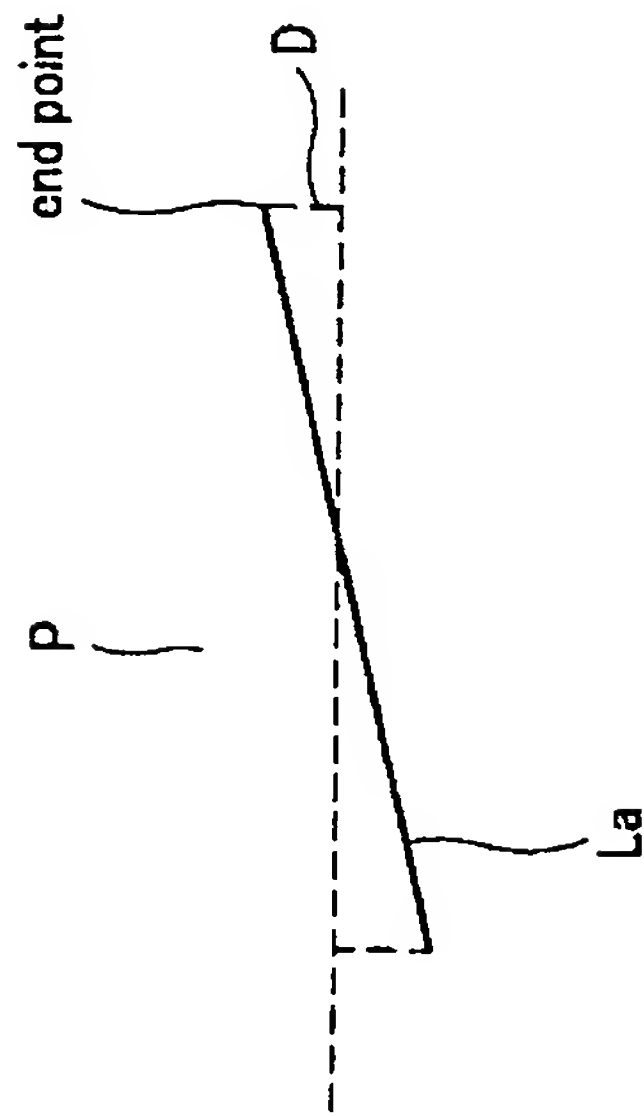


FIG.51A

[図52]

Algorithm *FindSeedRegion*Input: *lines[i]*: vector of lines for each image row (or column) *i*,*n*: number of image rows (or columns)

Output: set of lines (seed region) or empty set (no seed found).

Sequence:

for *i* = 0 to (*n*-3) do    for *l*<sub>1</sub> in *lines[i]* do        for *l*<sub>2</sub> in *lines[i+1]* do            for *l*<sub>3</sub> in *lines[i+2]* do                if *overlap(l*<sub>1</sub>, *l*<sub>2</sub>) and *overlap(l*<sub>2</sub>, *l*<sub>3</sub>) then                    (*n*, *d*) = *fitPlane(l*<sub>1</sub>, *l*<sub>2</sub>, *l*<sub>3</sub>)                    if *rms(l*<sub>1</sub>, *l*<sub>2</sub>, *l*<sub>3</sub>) < *threshl<sub>rms</sub>* then                        *seed* = {*l*<sub>1</sub>, *l*<sub>2</sub>, *l*<sub>3</sub>}                        *remove(l*<sub>1</sub>, *l*<sub>2</sub>, *l*<sub>3</sub>)                        return *seed*

endif

endif

endfor

endfor

endfor

endfor

return {}

FIG.52

[図53]

Algorithm *RegionGrowing*Input: *region* : set of lines as seed region,*lines[i]* : vector of lines for each image row (or column) *i*,*n* : number of image rows (or columns)

Sequence :

 $A = 0, b = 0$ for *l* in *region* do  $(A, b) = \text{add}(A, b, l)$  endfor $(n, d) = \text{solve}(A, b)$ *open* = *region*while not empty(*open*) do $l_1 = \text{select}(\text{open}), \text{open} = \text{open} - \{l_1\}$ for *i* in neighbor(index(*l*<sub>1</sub>)) dofor *l*<sub>2</sub> in *lines[i]* doif overlap(*l*<sub>1</sub>, *l*<sub>2</sub>) and  $\text{rms}(l_2) < \text{thresh2}_{\text{rms}}$  then $\text{region} = \text{region} + \{l_2\},$  $(A, b) = \text{add}(A, b, l_2), (n, d) = \text{solve}(A, b)$  $\text{open} = \text{open} + \{l_2\},$ remove(*l*<sub>2</sub>)

endif

endfor

endfor

endfor

*plane* = {*n, d, A, b, region*}*planes* = *planes* + {*plane*}

FIG.53

[図54]

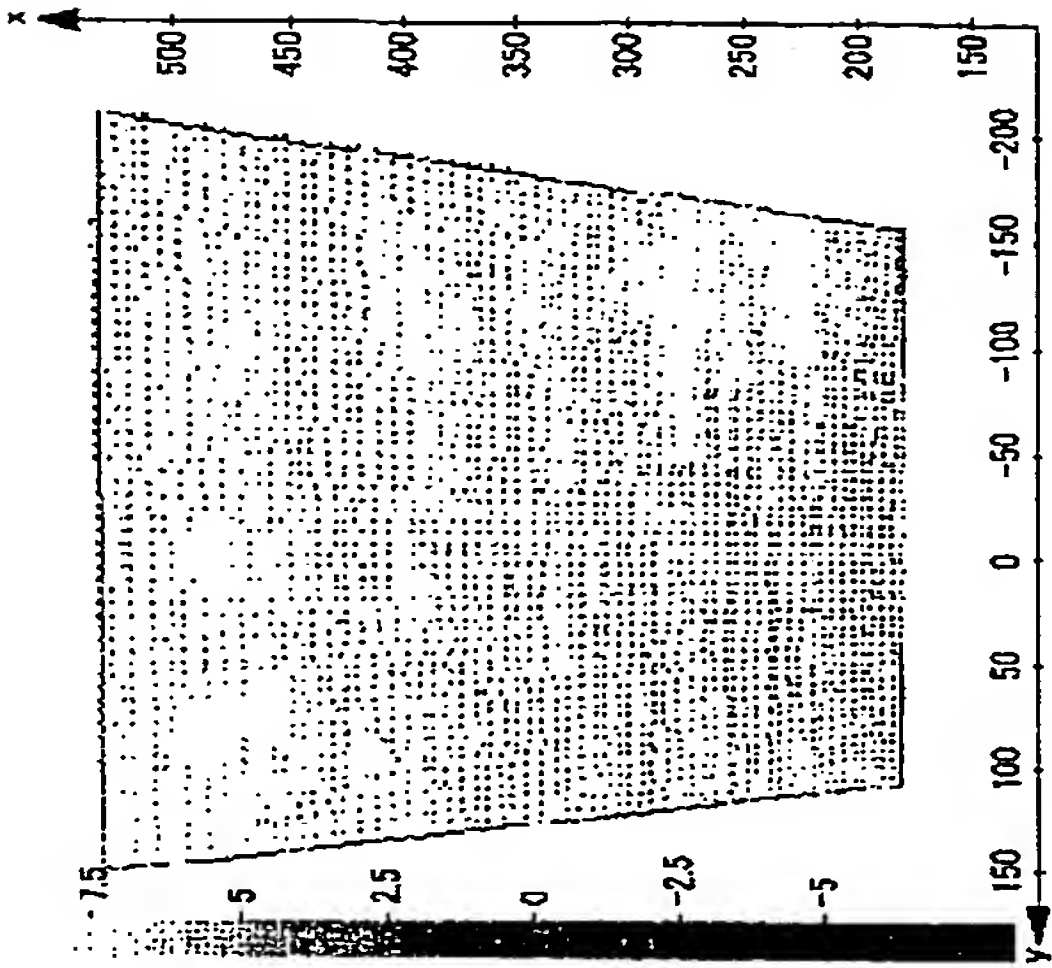


FIG. 54A

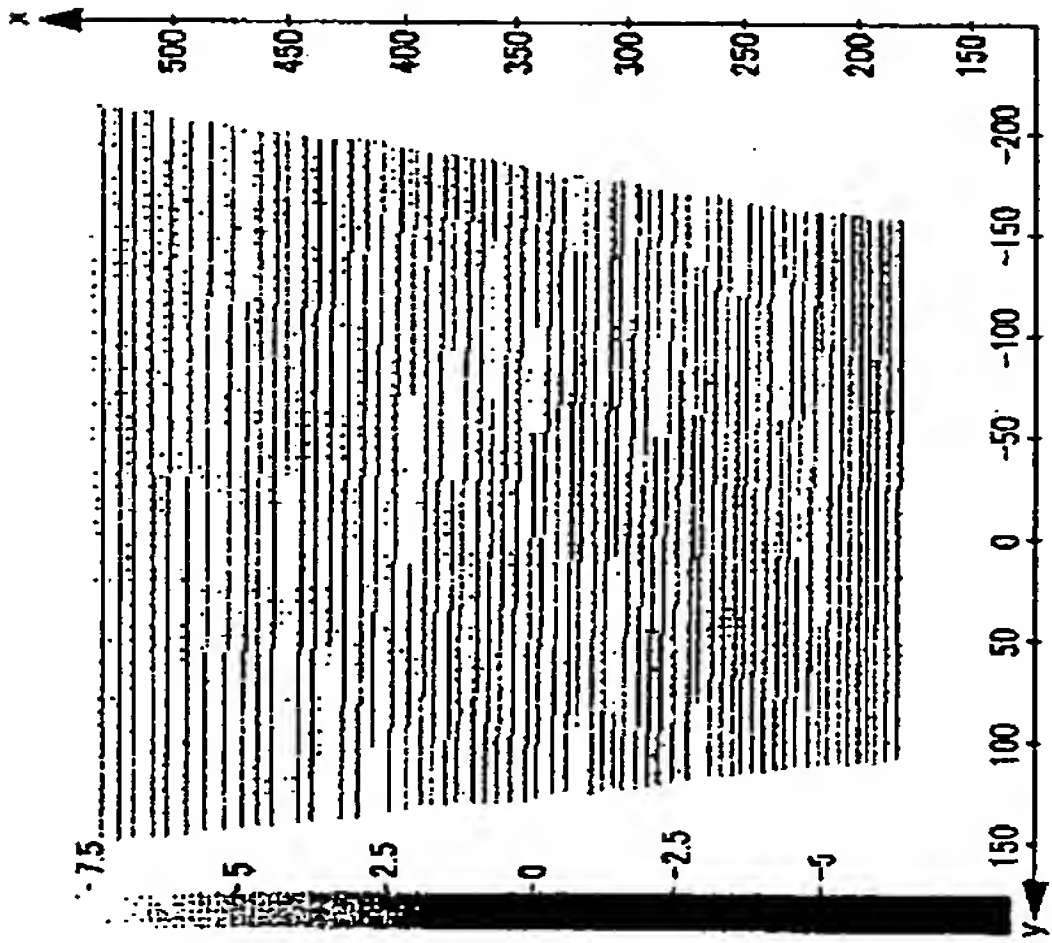


FIG. 54B

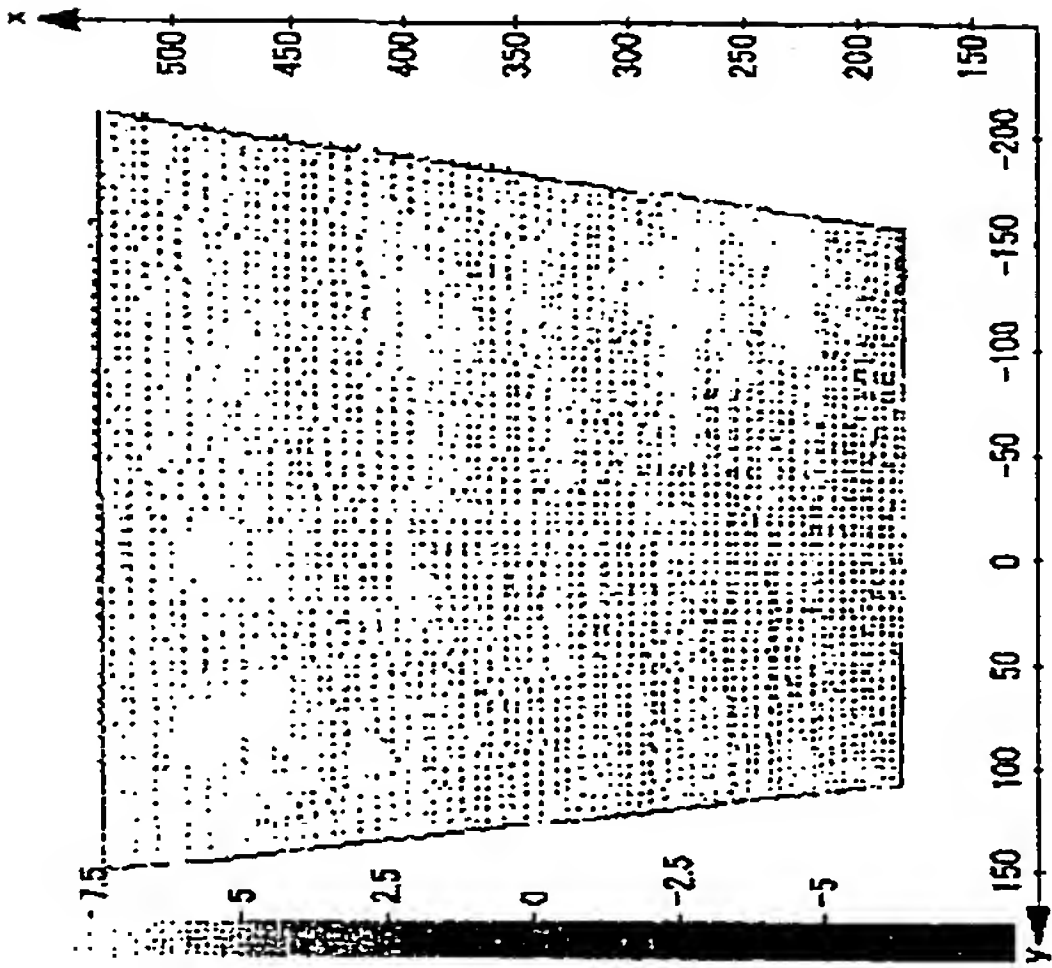


FIG. 54C

[図55]

No	max_d	enable zig-zag	correct extraction (horizontal)	correct extraction (vertical)
1	30	no	0 / 10	0 / 10
2	25	no	0 / 10	0 / 10
3	20	no	10 / 10	0 / 10
4	15	no	10 / 10	3 / 10
5	10	no	10 / 10	10 / 10
6	30	yes	10 / 10	10 / 10

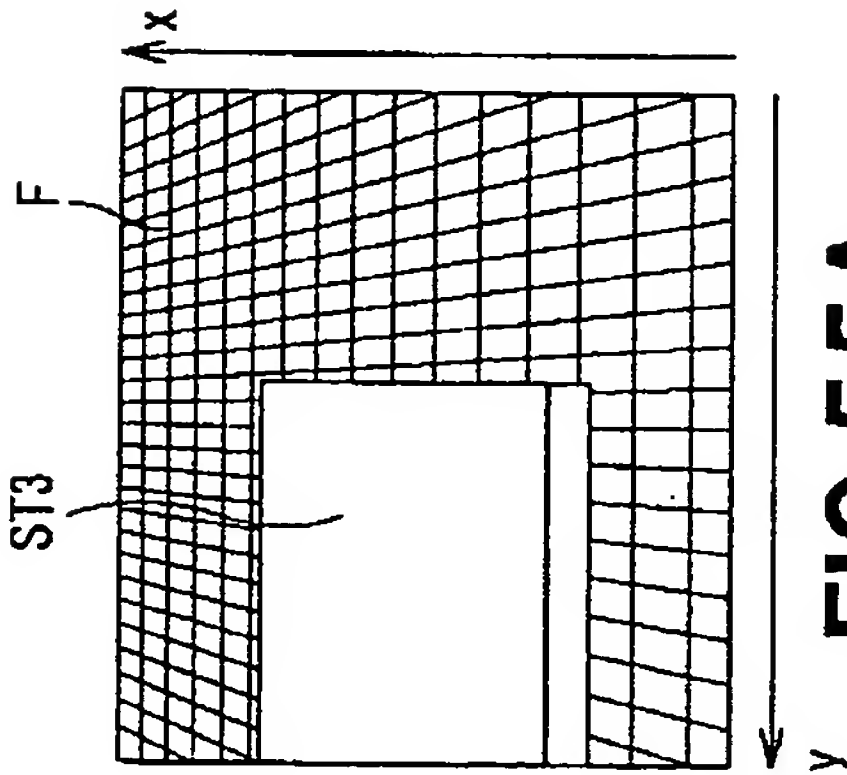


FIG. 55B

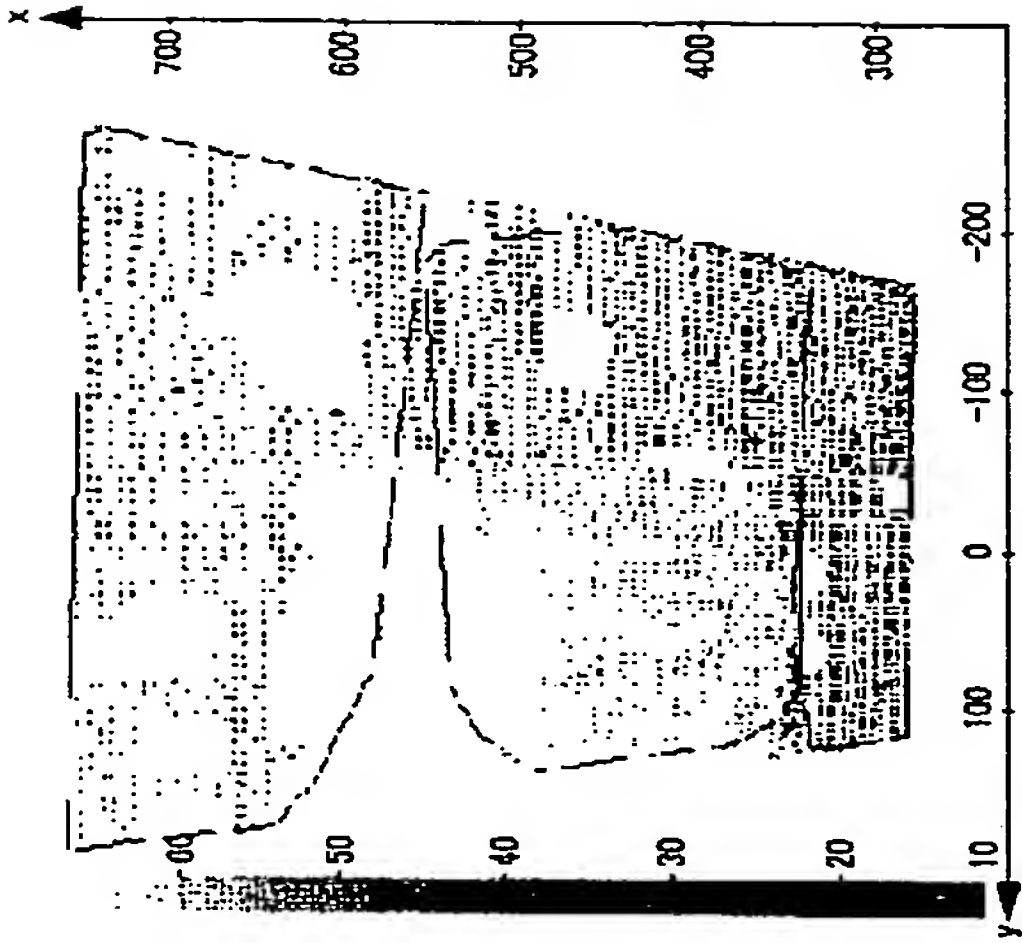


FIG. 55D

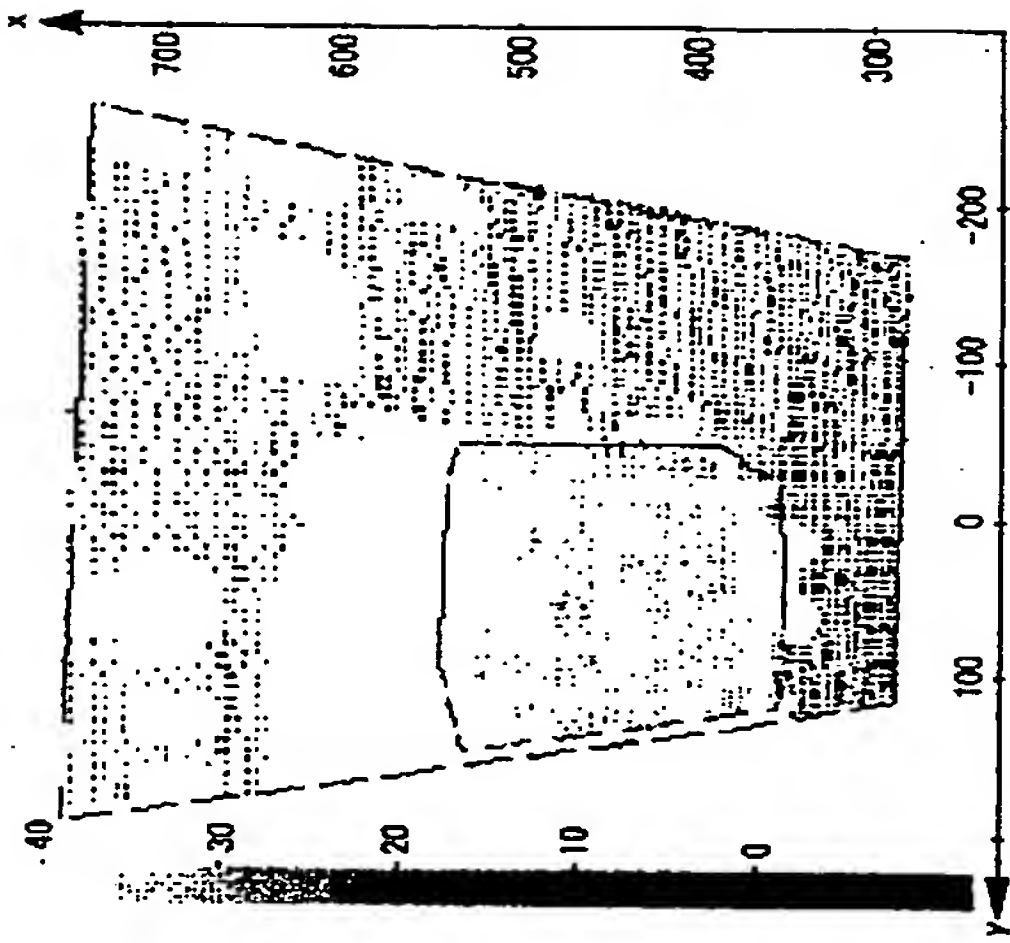


FIG. 55C



[図56]

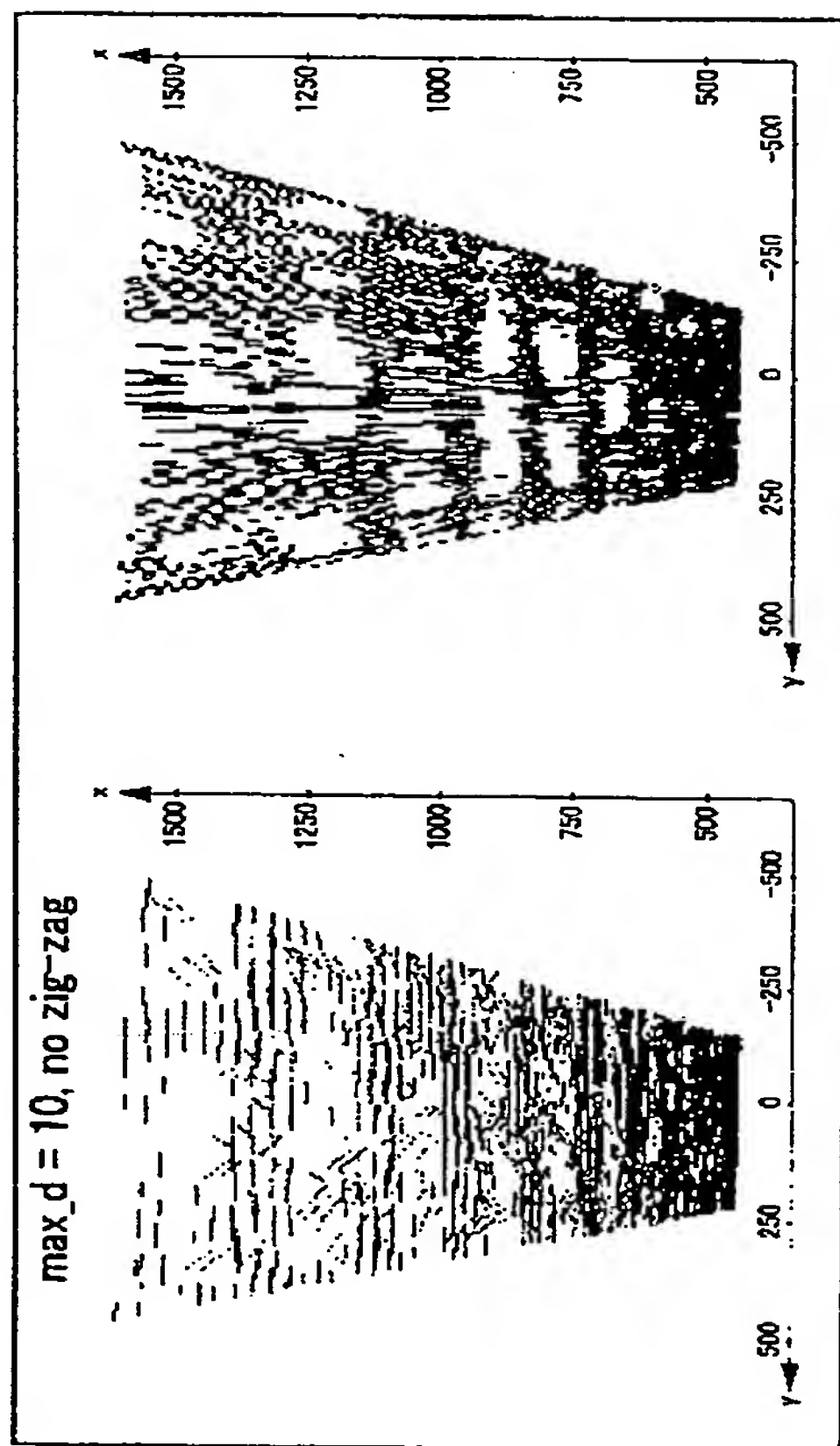


FIG. 56B

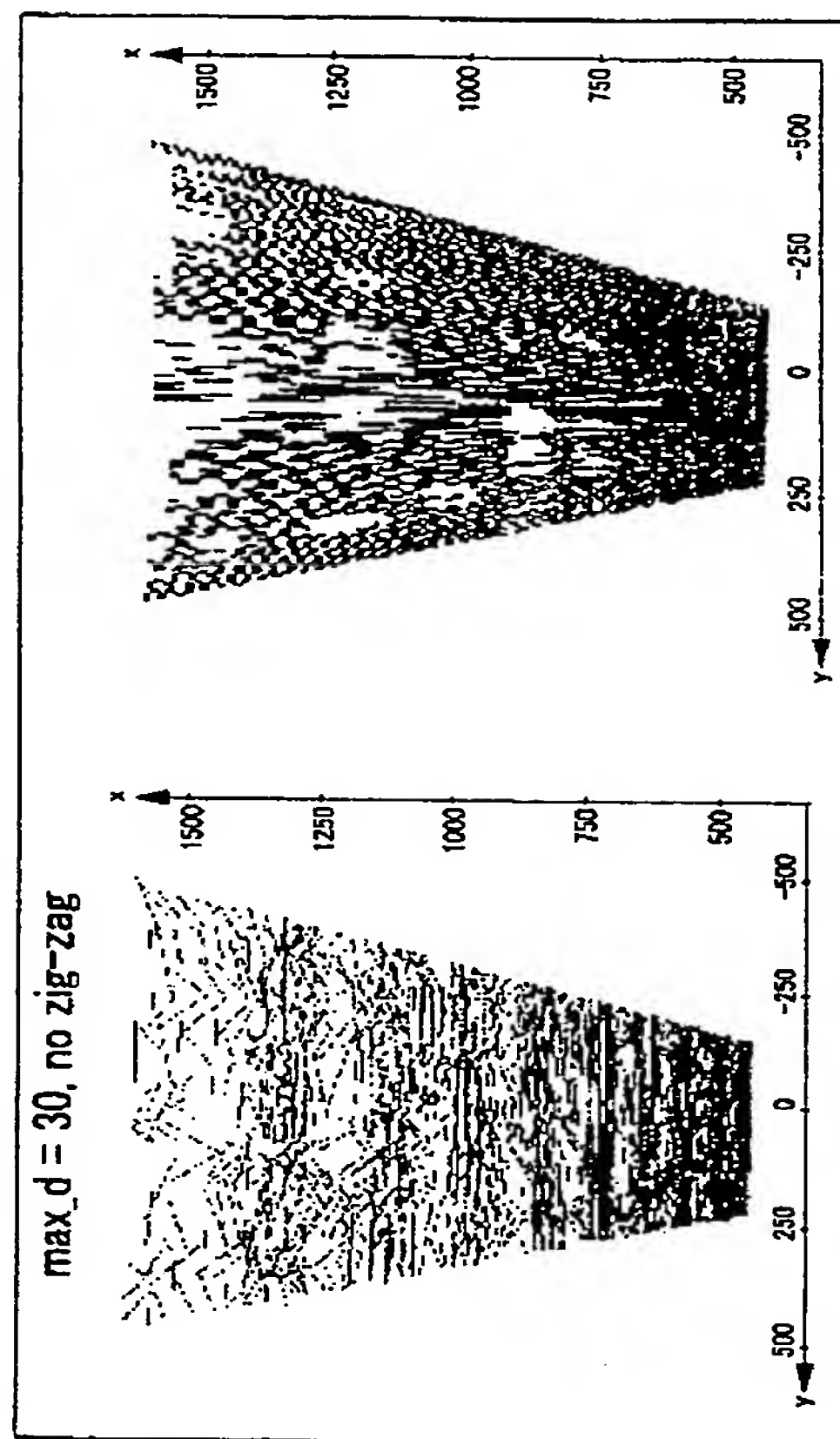


FIG. 56C

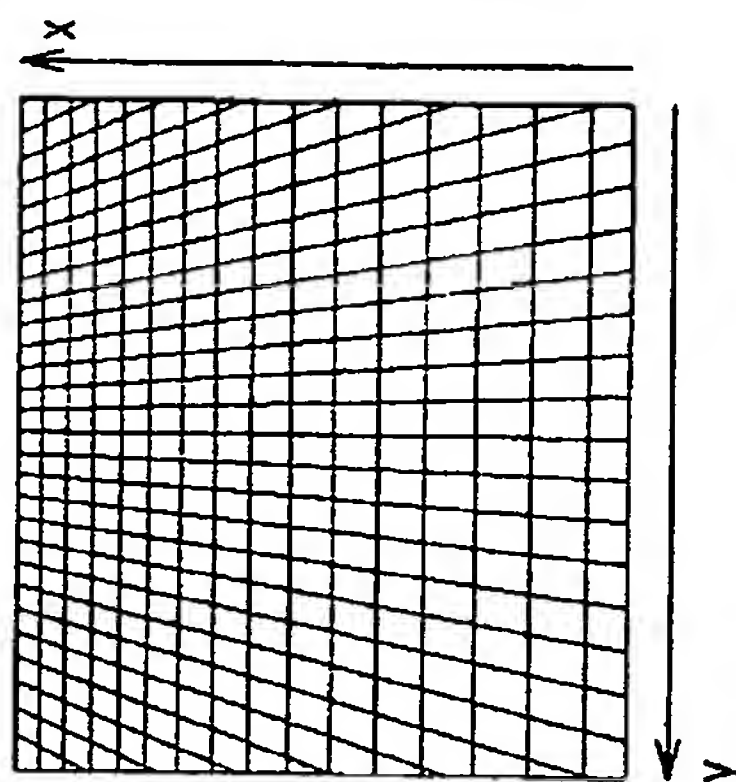


FIG. 56A

## INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2005/004838

## A. CLASSIFICATION OF SUBJECT MATTER

Int.Cl<sup>7</sup> B25J13/00, 5/00

According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

Int.Cl<sup>7</sup> B25J1/00-21/02

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Jitsuyo Shinan Koho	1922-1996	Jitsuyo Shinan Toroku Koho	1996-2005
Kokai Jitsuyo Shinan Koho	1971-2005	Toroku Jitsuyo Shinan Koho	1994-2005

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

IEEE, JSTPlus (JOIS), JST7580 (JOIS)

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	Sei OKADA et al., "Jitsujikan Plane Segment Finder no Kenkyu", Dai 6 Kai Robotics Symposia Yokoshu, 18 March, 2001 (18.03.01), pages 51 to 56	1-4, 6, 7, 12, 17, 19, 21-24, 32-35, 37, 39, 40, 42, 44-47
Y		9-11, 16, 38
A		5, 8, 13-15, 18, 20, 25-31, 36, 41, 43
Y	JP 5-223549 A (Honda Motor Co., Ltd.), 31 August, 1993 (31.08.93), Par. Nos. [0009], [0020] to [0023]; Fig. 32 (Family: none)	9-11, 16, 38
A		1-6, 8, 12-37, 39-47

☐ Further documents are listed in the continuation of Box C.☐ See patent family annex.

## \* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&amp;" document member of the same patent family

Date of the actual completion of the international search

30 June, 2005 (30.06.05)

Date of mailing of the international search report

19 July, 2005 (19.07.05)

Name and mailing address of the ISA/  
Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

A. 発明の属する分野の分類 (国際特許分類 (IPC)) Int.Cl. <sup>7</sup> B25J13/00, 5/00		
B. 調査を行った分野 調査を行った最小限資料 (国際特許分類 (IPC)) Int.Cl. <sup>7</sup> B25J1/00-21/02		
最小限資料以外の資料で調査を行った分野に含まれるもの 日本国実用新案公報 1922-1996年 日本国公開実用新案公報 1971-2005年 日本国実用新案登録公報 1996-2005年 日本国登録実用新案公報 1994-2005年		
国際調査で使用した電子データベース (データベースの名称、調査に使用した用語) IEEE, JSTPlus(JOIS), JST7580(JOIS)		
C. 関連すると認められる文献		
引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求の範囲の番号
X  Y A	岡田慧、外4名、実時間プレーンセグメントファインダの研究, 第6回ロボティクスシンポジウム予稿集, 2001/03/18, p. 51-56	1-4, 6, 7, 12, 17, 19, 21-24, 32-35, 37, 39, 40, 42, 44-47 9-11, 16, 38 5, 8, 13-15, 18 , 20, 25-31, 36 , 41, 43
<input checked="" type="checkbox"/> C欄の続きにも文献が列挙されている。 <input type="checkbox"/> パテントファミリーに関する別紙を参照。		
* 引用文献のカテゴリー 「A」 特に関連のある文献ではなく、一般的技術水準を示すもの 「E」 国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの 「L」 優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献 (理由を付す) 「O」 口頭による開示、使用、展示等に言及する文献 「P」 国際出願日前で、かつ優先権の主張の基礎となる出願日の後に公表された文献 「T」 国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの 「X」 特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの 「Y」 特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの 「&」 同一パテントファミリー文献		
国際調査を完了した日 30.06.2005	国際調査報告の発送日 19.7.2005	
国際調査機関の名称及びあて先 日本国特許庁 (ISA/JP) 郵便番号100-8915 東京都千代田区霞が関三丁目4番3号	特許庁審査官 (権限のある職員) 八木 誠 電話番号 03-3581-1101 内線 3324	3U 3504

C (続き) . 関連すると認められる文献		
引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求の範囲の番号
Y A	JP 5-223549 A (本田技研工業株式会社) 1993. 08. 31, 段落【0009】, 【0020】 - 【0023】, 図 32 (ファミリーなし)	9-11, 16, 38 1-6, 8, 12-37, 39-47